UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Douglas Nazareno Debiazi Vargas

UMA FERRAMENTA PARA GERÊNCIA DE REDES ATM, VIA WWW, JAVA E SNMP

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Prof. Dr. Bernardo Gonçalves Riso
(Orientador)

Prof. Dr. Carlos Becker Westphall
(Co-orientador)

Florianópolis, fevereiro de 2001.

UMA FERRAMENTA PARA GERÊNCIA DE REDES ATM, VIA WWW, JAVA E SNMP

por

Douglas Nazareno Debiazi Vargas

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração: Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Prof. Dr. Fernando Augusto Ostuni Gauthier
Coordenador de Curso

Banca Examinadora

Bernardo Gonçalves Riso

Prof. Dr. Bernardo Gonçalves Riso
Orientador

Prof. Dr. Carlos Becker Westphall
Co-orientador

Prof. Dr. Fernando Augusto Ostuni Gauthier

Profa. Dra. Mirela Sechi Moretti Annoni Notare

Dedicatória

Dedico esta dissertação aos meus pais e à minha família, pois cada um deles ao seu modo me incentivou e contribuiu para a realização deste trabalho.

E à Gisele, minha noiva, pelo incentivo, pelo carinho e pela compreensão que teve, em todos os momentos nos quais a dedicação aos trabalhos se interpôs a nós.

AGRADECIMENTOS

Com estima e apreço agradeço aos professores e amigos Prof. Dr. Bernardo Gonçalves Riso e Prof. Dr. Carlos Becker Westphall, que disponibilizaram seu tempo para as discussões, os conselhos e as orientações, num convívio engrandecedor tanto acadêmico como pessoal.

Estendo os agradecimentos também para todos os demais professores do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina, os quais me permitiram durante a convivência conceber conceitos e adquirir conhecimentos que contribuíram para a realização deste trabalho.

Ao pessoal do NPD: André, Jussara, Fernando e Elvis, agradeço pela paciência, pelo suporte e pelas conversas enriquecedoras, em especial a Solange Sari pelo auxílio nas re-configurações dos *switch's* e ao Edison Melo que além de tudo foi um incentivador desde o início do trabalho.

A Vera e a Valdete, que atuando pela secretaria do Curso de Pós-Graduação em Ciência da Computação, sempre demonstraram presteza e zelo ao cuidar de todos os aspectos relacionados aos interesses dos alunos.

Aos colegas de curso e de sala de estudo, em especial aos amigos José Luis, Gilmário e Ewerton, pela amizade e pelo companheirismo, mostrado durante todo o trabalho, em especial durante a realização das disciplinas.

A Deus...

SUMÁRIO

LISTA DE ABREVIATURAS

AAL	*ATM Adaptation Layer*
ACAFE	Associação Catarinense das Fundações Educacionais
ATDM	*Asynchronous Time Division Multiplexing*
ATM	*Asynchronous Transfer Mode*
B-ISDN	*Broadband Integrated Services Digital Networks*
CCITT	*Comité Consultatif International Télégraphique et Téléphonique*
CE	Controladores de Entrada
CFM	Centro de Ciências Físicas e Matemáticas
CGI	*Common Gateway Interface*
CLIMERH	Centro Integrado de Meteorologia e Recursos Hídricos de Santa Catarina
CLP	*Cell Loss Priority*
CMIP	*Common Management Information Protocol*
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
COPPE	Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia
CPGCC	Curso de Pós-Graduação em Ciência da Computação
CS	Controladores de Saída
CS	*Convergence Sub-layer*
CTC	Centro Tecnológico
EPAGRI	Empresa de Pesquisa e Agropecuária e Difusão de Tecnologia de Santa Catarina
FAPERJ	Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro
FAPESP	Fundação de Amparo à Pesquisa do Estado de São Paulo
FAPERGS	Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul
FDM	*Frequency Division Multiplexing*
FIESC	Federação das Indústrias do Estado de Santa Catarina
FINEP	Financiadora de Estudos e Projetos
FTP	*File Transfer Protocol*

FUNCITEC	Fundação de Ciência e Tecnologia
GFC	*Generic Flow Control*
HEC	*Header Error Check*
HTML	*Hypertext Markup Language*
HTTP	*Hypertext Transfer Protocol*
FDDI	*Fiber Distributed Data Interface*
IEEE	*Institute of Electrical and Electronics Engineers*
IETF	*Internet Engineering Task Force*
ILMI	*Interim Local Management Interface*
INE	Departamento de Informática e de Estatística
ISDN	*Integrated Services Digital Network*
ITU	*International Telecommunications Union*
LAN	*Local Area Network*
LB	Ligação Básica
LMP	Ligação Multiplexada Primária
LTT	*Local Translation Table*
MAN	*Metropolitan Area Network*
MCT	Ministério da Ciência e Tecnologia
MIB	*Management Information Base*
MP	Módulos Processadores
N-ISDN	*Narrowband Integrated Services Digital Networks*
NNTP	*Network News Transfer Protocol*
NNI	*Network-Network Interface*
NPD	Núcleo de Processamento de Dados
NT	*Network Termination*
OAM	*Operation Administration and Maintenance*
PCM	*Pulse Code Modulation*
PM	*Physical Medium*
PoP	Ponto de Presença
PoP-SC	Ponto de Presença de Santa Catarina
PRM	*Protocol Reference Model*

PVC	*Permanent Virtual Connections*
ProTeM-CC	Programa Temático Multi-Institucional em Ciência da Computação
PTI	*Payload Type Identifier*
PUC-RIO	Pontifícia Universidade Católica do Rio de Janeiro
QoS	*Quality of Service*
RCT	Rede de Ciência e Tecnologia
RCT-SC	Rede de Ciência e Tecnologia de Santa Catarina
RDSI	Redes Digitais de Serviços Integrados
RDSI-FE	Rede Digital de Serviços Integrados de Faixa Estreita
RDSI-FL	Rede Digital de Serviços Integrados de Faixa Larga
RENAV	Rede Nacional de Alta Velocidade
RENPAC	Rede Nacional de Pacotes
RMAV-FLN	Rede Metropolitana de Alta Velocidade de Florianópolis
RNP	Rede Nacional de Pesquisa
SAR	*Segmentation and Reassembly*
SAS	*SNMP Applet Server*
SAAL	*Signaling AAL*
SC	*Technical Sub-committees*
SDH	*Synchronous Digital Hierarchy*
SEBRAE	Serviço de Apoio à Pequena e Média Empresa
SIGMA/WS	Ferramententa para a Gerência e a Monitoração de redes ATM via *Web* com SNMP
SMTP	*Simple Mail Transfer Protocol*
SNMP	*Simple Network Management Protocol*
SONET	*Synchronous Optical Network*
STDM	*Synchronous Time Division Multiplexing*
STM	*Synchronous Transfer Mode*
SVC	*Switched Virtual Connections*
SVCC	*Signaling Virtual Channel Connection*
TC	*Technical Committees*
TC	*Transmission Convergence*

TCP/IP	*Transfer Control Protocol/Internet Protocol*
TDM	*Time Division Multiplexing*
TELESC	Empresa de Telecomunicações de Santa Catarina
UDESC	Universidade do Estado de Santa Catarina
UFSC	Universidade Federal de Santa Catarina
UME	*UNI Management Entities*
UNI	*User-Network Interface*
VCC	*Virtual Channel Connection*
VCI	*Virtual Channel Identifier*
VCL	*Virtual Channel Link*
VPC	*Virtual Path Connection*
VPI	*Virtual Path Identifier*
VPL	*Virtual Path Link*
WAN	*Wide Area Network*
WBM	*Web-Based Management*
WG	*Working Group*
WWW	*World Wide Web*

LISTA DE FIGURAS

Lista de Tabelas

RESUMO

Este trabalho descreve estudos realizados sobre: as redes ATM, a gerência dessas redes com uso de SNMP e da Internet, com maior atenção ao ambiente WWW e à linguagem Java. Estas tecnologias servem como base para o desenvolvimento de uma ferramenta de gerência de redes ATM, a ferramenta SIGMA/WS, com a qual foram realizados experimentos no ambiente de rede ATM da UFSC. A SIGMA/WS foi desenvolvida tendo como base a linguagem HTML e a linguagem Java, através das quais, estabelece-se uma comunicação via SNMP com os equipamentos de rede gerenciados. Para fazer funcionar a ferramenta SIGMA/WS faz-se necessário dispor de um cliente *Web* instalado em um computador conectado à Internet. A ferramenta SIGMA/WS foi empregada em um ambiente de teste para monitorar um *switch* ATM da rede UFSC. Nesse ambiente, foi monitorado o tráfego dos enlaces conectados a esse *switch*, obtendo-se gráficos que apresentam informações relacionadas à vazão, volume e qualidade de serviço dos enlaces, além de informações sobre a configuração do *switch*. Através desses gráficos pode-se analisar o comportamento da rede, verificando os enlaces com gargalo e horários de pique. Como resultado global, este trabalho mostra o uso do ambiente *Web* para a monitoração de aspectos de desempenho, de configuração e falhas aplicado a redes ATM, com o uso do SNMP, graças ao emprego de uma ferramenta desenvolvida com as linguagens HTML e Java.

ABSTRACT

This work describes studies developed through on: networks ATM, manage it of these networks with use of SNMP and the Internet, with greater attention to the WWW environment and the Java language. These technologies serve as base for developing a networks ATM tool manages, the tool SIGMA/WS, with which experiments in the environment of network ATM of the UFSC had been carried through. The SIGMA/WS was developed having as base the language HTML and the Java language, through which, establishes a communication by SNMP with the managed equipment of network. To make to work the tool SIGMA/WS becomes necessary to make use of a Web-Client installed in a computer hardwired to the Internet. Tool SIGMA/WS was used in a test environment to monitor one switch ATM of network UFSC. In this environment, the traffic of the links connected to this switch was monitored, getting itself graphical that present information related to the outflow, volume, quality of job of link them, beyond information on the pattern switch. The analysis of these graphs allows to evidence the behavior of the network, denouncing you link with pass and schedules of puncture. As global result, this work shows the viability of the use of the Web environment for the monitoration of performance aspects and configuration applied to the networks ATM, with the use of the SNMP, favours to the use of a tool developed with languages HTML and Java.

Capítulo 01: Introdução

Capítulo 01: Introdução

Desde os primórdios, a humanidade vem evoluindo na busca contínua de melhorias de vida e de bem estar. Nesse processo, a sociedade vem modificando a sua forma de organização e de relacionamento, ou seja, adaptando sua cultura a cada nova invenção ou tecnologia desenvolvidas pelo homem.

As novas tecnologias surgem como uma resposta de pesquisa e desenvolvimento científico frente a um anseio da sociedade. Após a sua implantação, porém, a tecnologia desenvolve-se de acordo com a sua própria lógica, podendo fazer com que o indivíduo sinta-se impotente frente às inovações tecnológicas e às conseqüentes mudanças na vida social, política e econômica, que elas implicam.

A ciência determina novos paradigmas, os quais são utilizados pelas indústrias para o desenvolvimento de novos produtos, e a sociedade define o seu uso. Nesse processo, a sociedade adapta-se às novas tecnologias ao incorporar seus produtos e serviços no cotidiano. A Internet é um exemplo real de uma tecnologia que foi desenvolvida com uma finalidade bem específica (projeto militar); disseminada com outra (finalidade educacional) e para a qual a sociedade criou usos e aplicações que não foram totalmente previstos inicialmente (diversão, comércio, publicidade, etc.).

Conforme Paul Romer (ROMER, 1996), "O mito do vale do silício começa com a descoberta do transistor. Desdobrando-se de acordo com a lógica irreversível da Lei de Moore". Frederico Faggin (FAGGIN, 1996) constata, "Com a invenção do circuito integrado, o número de transistores contido em um *chip* aumentou um milhão de vezes".

Os mitos se formam pelo impacto que eles causam em nosso dia-a-dia e não é exagero colocar a invenção do transistor e o seu subseqüente aperfeiçoamento como fator principal para o início de uma revolução. O mito do transistor nos trouxe: a informatização, a automação, a globalização, a economia baseada no conhecimento, enfim, a Revolução Digital.

O motivo pelo qual se considera que a invenção do transistor tenha sido um marco tão importante na história, pode ser creditada a diversos fatos:

- os princípios básicos de uma máquina de cálculo, capaz de efetuar uma ampla gama de cálculos de acordo com as instruções oferecidas por seu operador, foram concebidos em 1840 por Charles Babbage, norteando a teoria dos computadores programáveis de hoje;
- por volta de 1900, já eram conhecidos os princípios do armazenamento magnético;
- poucos anos depois, George Boole, estudando a lógica binária, concebeu um sistema algébrico de símbolos e regras, chamado de lógica booleana, introduzindo também o conceito de portas lógicas;
- em 1936, Claude Shannon, matemático, e engenheiro eletricista, percebendo a relação entre a Lógica Booleana e os Circuitos Elétricos, estudou a relação entre a teoria algébrica e a aplicação prática para implementar um circuito somador elétrico;

Portanto, cem anos se passaram para que o computador concebido por Babbage pudesse ser construído. Ao longo desses anos, tecnologias foram desenvolvidas, teorias concebidas, equipamentos inventados, e idéias foram adaptadas, até que estivesse disponível a tecnologia suficiente para que o computador fosse construído.

A constatação de Paul Romer (ROMER, 1996), nos demonstra a importância do transistor: "Antes do transistor, existiam poucos computadores e após a invenção do transistor, o poder de processamento do computador aumentou rapidamente, transformando as nossas vidas".

Por ser um equipamento multipropósito, programável – podendo assim ser utilizado nas mais diversas áreas – o computador provocou um impacto muito grande na sociedade, revolucionando irreversivelmente os processos conhecidos.

A programabilidade, concebida por Charles Babbage, é a característica chave que torna o computador um equipamento versátil, capaz de executar qualquer tarefa programável. Complementarmente, a flexibilidade de seu interfaceamento com outros dispositivos permite o uso e o auxílio do computador em qualquer tipo de tarefa.

A aplicação cada vez maior da informática tem contribuído para a evolução da sociedade com a introdução de diversas tecnologias, aperfeiçoando e criando processos que transformam os nossos costumes. Atualmente, qualquer inovação tecnológica em computadores causa um impacto muito grande junto à sociedade, devido ao alto nível de informatização, hoje alcançado em praticamente todos os ramos de atividade.

A tecnologia das redes de computadores, por sua vez não foge à regra, principalmente porque esta é a tecnologia responsável pela interconexão dos computadores, o que multiplica o seu impacto junto à sociedade. A Internet, sendo a tecnologia de rede mais utilizada atualmente, serve como exemplo desse impacto junto à sociedade. O avanço na globalização, a facilidade das transmissões, o surgimento do comércio eletrônico e o crescimento das organizações virtuais, por exemplo, podem ser atribuídos, em grande parte, ao seu uso.

Diversas tecnologias foram paralelamente implementadas, como aperfeiçoamento e também como um complemento à tecnologia de redes, para que se pudesse chegar ao nível tecnológico atual na interconexão dos computadores. Dos equipamentos até a interface com o usuário – passando por várias camadas de protocolos, até a camada de aplicação, muitos aperfeiçoamentos foram implementados em: tecnologias de transmissão, equipamentos de rede, protocolos de comunicação, protocolos de serviços, aplicações de rede e seus protocolos.

Segundo Luiz Alves (ALVES, 1994), o processo de transmissão envolve cinco partes: o transmissor; a mensagem; o canal de comunicação; o protocolo de comunicação e o receptor.

Para trocar informações (mensagens) entre computadores (emissor e receptor), deve-se ter um meio físico (canal de comunicação), porém, a troca de informações torna-se viável a partir da definição de um conjunto de regras (protocolo de comunicação) que tratam das mais diversas situações (erros, impasses, falhas, etc.), e permitem que a comunicação se efetue.

Partindo da camada física, ou seja, no nível de cabos e equipamentos de rede, pode-se dizer que o que ocorre entre os computadores é a troca de *bits*. Os *bits* são os sinais (elétricos, ópticos, microondas, radiofreqüências, infravermelhos, etc.) que transportam, agrupados em *bytes*, as informações trocadas.

O **protocolo de comunicação** é o conjunto de regras que define como se dá a troca de informação. Desde a localização do computador de destino na rede, o estabelecimento de uma conexão entre os computadores, a troca de informações entre eles, o tratamento de erros, até a finalização da conexão. Na Internet, o conjunto de protocolos TCP/IP *(Transfer Control Protocol/Internet Protocol)* é responsável pelo estabelecimento e a troca de informações entre os computadores no nível do protocolo de comunicação.

Acima dos protocolos de comunicação e utilizando-se dos serviços que eles disponibilizam, estão os **protocolos de serviços**, que são responsáveis pelos serviços básicos providos pela rede. Os serviços que a rede disponibiliza, formam os recursos que são utilizados pelos usuários, e tornam-se, assim, a porção da rede que é visivelmente utilizada por eles.

A Internet utiliza-se de um grupo de protocolos para fornecer um conjunto básico de serviços aos usuários da rede. Estes serviços são disponibilizados por seus respectivos protocolos de serviços. A saber: o serviço de correio eletrônico usa o protocolo SMTP (*Simple Mail Transfer Protocol*); o serviço de transmissão de arquivos utiliza-se do protocolo FTP (*File Transfer Protocol*); o serviço de grupo de notícias ou discussão usa o protocolo NNTP (*Network News Transfer Protocol*); e o serviço de

navegação por hipertexto ou WWW utiliza-se do protoloco HTTP (*Hypertext Transfer Protocol*).

Existem ainda, os chamados **protocolos de aplicação**, que são utilizados por uma aplicação específica. Eles definem como duas aplicações ou duas instâncias de uma mesma aplicação podem trocar informações, remotamente, através da rede (usando o protocolo de comunicação da rede). Na Internet, o Telnet serve como exemplo de uma aplicação que usa o seu próprio protocolo de aplicação para estabelecer a comunicação cliente/servidor, remotamente, sobre o TCP/IP.

Os inúmeros serviços disponibilizados por todos os protocolos de uma rede, e as aplicações que os utilizam é o que vem transformando a sociedade atual, através da quantidade cada vez maior de recursos oferecidos. Estes recursos podem ser traduzidos em: mais facilidade, melhor performance, maior segurança e compatibilidade, enfim, atrativos que nos fazem querer usar, cada vez mais, a tecnologia de rede.

Existe porém, um serviço de rede que não é focado diretamente para o usuário. Trata-se do serviço de gerenciamento de rede. Ele utiliza-se de um protocolo desenvolvido com uma conotação diferenciada de todos os outros. O **protocolo de gerência de rede** foi concebido para auxiliar a administração da rede. Esse protocolo tem por objetivo obter informações sobre a rede, monitorá-la e mantê-la em funcionamento para que todos os outros serviços da rede permaneçam disponíveis. O protocolo de gerência de rede nativamente utilizado na Internet é o SNMP (*Simple Network Management Protocol*).

O atual estágio de difusão das redes de computadores – alavancado nos últimos anos pela Internet – assim como o seu uso para aplicações cada vez mais críticas, justifica o estudo, a pesquisa e o desenvolvimento de tecnologias que aperfeiçoem as redes de computadores. É dentro deste escopo que se enquadra a pesquisa e o desenvolvimento em gerência de rede.

A gerência de rede é uma forma através da qual a rede é monitorada com o objetivo de: melhorar a performance, descobrir falhas, diminuir erros e retransmissões, enfim, manter a rede ativa.

1.1 Objetivo

O objetivo deste trabalho é desenvolver uma ferramenta de gerência de redes baseada em um ambiente *Web*, que monitore uma rede ATM, através do protocolo SNMP com o uso da linguagem Java. Mostrando que a flexibilidade, a versatilidade e a eficiência do ambiente hipertexto *Web* constituem um ambiente adequado para a execução de uma ferramenta de gerência de redes ATM.

1.2 Método

Para atingir tal objetivo estudou-se a tecnologia de redes de computadores e a gerência de redes em especial, sendo aprofundado o estudo nas tecnologias de redes de alta velocidade, mais especificamente, ATM.

Considerando o crescimento da Internet e a sua importância no contexto das redes de computadores, foram estudadas as tecnologias a ela incorporadas, para que fosse possível estabelecer quais os recursos que deveriam ser utilizados.

Após a definição das tecnologias e recursos utilizados, deu-se início à implementação de uma ferramenta (SIGMA/WS) que se comunica com os equipamentos gerenciáveis da rede para obter informações de gerenciamento. Estas informações estão contidas na Base de Informações de Gerenciamento (ou *Management Information Base* – MIB) do equipamento.

Terminada a implementação, a ferramenta desenvolvida é utilizada para monitorar o ambiente de teste estabelecido, o qual é composto por um *switch* ATM e seus enlaces. A monitoração do ambiente de teste visa verificar a utilidade e a qualidade das

informações disponibilizadas pela ferramenta, assim como a sua aplicabilidade para as atividades de monitoração e de gerência, validando o uso da ferramenta (SIGMA/WS).

Para efetuar os testes e os experimentos com a ferramenta, fez-se necessário um computador conectado à Internet – com número de IP fixo – de tal modo que, através deste endereço o mesmo pudesse ser identificado pelo equipamento gerenciável (*switch*), como membro legítimo de uma *community*, permitindo assim o seu acesso aos dados da MIB.

Uma vez que se tinha um computador com acesso disponível ao elemento gerenciável, era necessário que o mesmo executasse um *Web Server* para que a ferramenta desenvolvida em páginas HTML pudesse ser visitada. Além disso, era necessário que este computador tivesse disponível um *Applet Server* para executar os *Applet's* contidos nas páginas da ferramenta.

Assim sendo, durante a fase de testes iniciais foi utilizado um computador da sala de estudos do CPGCC para gerenciar o ambiente de teste definido, posteriormente, isto foi feito de fora da RedeUFSC, de um computador localizado na cidade de Lages.

1.3 Justificativa

As redes de computadores têm um papel cada vez mais importante no cotidiano de todos, sendo que, ao ampliarmos os seus usos, torna-se premente a melhoria na sua monitoração, na facilidade de administrá-la, na garantia de seus serviços e na sua disponibilidade. Todos estes recursos devem ser oferecidos ou facilitados por uma ferramenta de gerência de rede.

1.4 Organização do Trabalho

Este trabalho está organizado da seguinte maneira, no Capítulo 2, apresenta-se a tecnologia de rede ATM, definindo-a como uma rede de comutação e multiplexação de

pacotes, que constitui uma Rede Digital de Serviços Integrados. Apresenta-se, ainda o modelo de referência para RDSI-FL e são detalhadas as camadas (Física, ATM e de Adaptação), assim como os planos (de Usuário, de Controle e de Gerência). Finalizando, discorre-se sobre a operação de uma rede ATM.

A gerência das redes de computadores é o assunto do Capítulo 3, no qual, após justificar a necessidade e o estudo dessa gerência, apresenta-se o uso das tecnologias WWW, SNMP e Java para gerenciar redes, terminando com uma apresentação da ferramenta Management Builder da Adventnet.

No Capítulo 4, discorre-se sobre o ambiente de experimentação e teste. Nesses capítulos é apresentada a instituição (UFSC), o departamento (INE), e o curso (CPGCC) no qual esta dissertação foi desenvolvida, e o NPD da UFSC. Para tanto, um histórico é mostrado, destacando o ambiente computacional da UFSC com sua rede e suas interconexões (RNP, RCT, Embratel, TELESC, RMAV-FLN e redeUFSC).

O SIGMA/WS, Sistema de Informações para a Gerência e a Monitoração de redes ATM, via *Web* com SNMP, é apresentado no Capítulo 5. Suas características são detalhadas e seus recursos apresentados, assim como a sua forma de desenvolvimento, além de justificado o seu uso.

Os experimentos realizados são relatados no Capítulo 6. No Capítulo 7 apresenta-se a conclusão, onde também se discorre sobre a interpretação e a análise dos resultados e os trabalhos futuros. Seguem-se os Anexos e a Bibliografia.

Capítulo 02: Redes ATM

Capítulo 02: Redes ATM

Uma rede de computadores é, segundo Tanenbaum (TANENBAUM, 1997) "**um conjunto de computadores autônomos interconectados**", independentemente de como a conexão entre os computadores seja feita. Assim sendo, uma vez que existam dois computadores interconectados, e entre eles haja a disponibilidade de comunicação, entende-se que estes computadores estão conectados através de uma rede.

Abstraindo detalhes técnicos, pode-se dizer que a estrutura básica de uma rede de computadores é formada por: dois ou mais computadores (*hosts* conforme Tanenbaum (TANENBAUM, 1997) ou **módulos processadores MP** conforme Soares (SOARES, 1997)) interconectados através de uma sub-rede (**sistema de comunicação**, conforme Soares (SOARES, 1997), ou uma **sub-rede de comunicação** conforme Tanenbaum (TANENBAUM, 1997)). Ver Figura 1.

Figura 1: A estrutura de uma rede de computadores.
Fonte: (TANENBAUM, 1997). Pág. 6.

A sub-rede de comunicação, ou simplesmente sub-rede, viabiliza a comunicação entre os computadores na medida em que é responsável pelo transporte de dados entre eles.

As redes de computadores – como infra-estruturas que permitem aos computadores compartilharem recursos, dados e aplicações entre si – são vinculadas a muitos usos dos computadores. Deve, entretanto, permitir que os recursos possam ser compartilhados adequadamente, ampliando assim sua capacidade de tráfego e permitindo garantir a disponibilidade e a qualidade de serviços.

Nos últimos anos, porém, o uso de redes de computadores deixou de ser uma aspiração corporativa e tornou-se também uma aspiração pessoal. Os principais motivos que hoje levam as pessoas a interessarem-se por uma rede são: ter rápido acesso às informações; comunicar-se interpessoalmente; executar procedimentos remotos e diversão interativa (TANENBAUM, 1997).

O real uso deste tipo de tecnologia tornou-se viável somente a partir da disseminação do uso da rede Internet. Não se pode esquecer porém que, quanto mais usuários houver em uma rede, maior será o seu tráfego, e conseqüentemente a demanda desses usuários por uma maior velocidade.

As Redes de Alta Velocidade

A evolução nas redes de computadores tem acontecido nas mais diversas áreas de aplicação, para ampliar a compatibilidade, a segurança, a gerência, e a performance, entre outros avanços. O objetivo desses avanços, é sempre permitir que as redes suportem aplicações cada vez mais sofisticadas.

A definição do que venha a ser uma aplicação sofisticada, não é bem estabelecida, porém, no contexto atual, refere-se a aplicações ditas no estado-da-arte, e que façam uso de: multimídia (imagens e sons); processamento e/ou armazenamento distribuídos; sistemas em tempo real; etc. (ALVES, 1994)

Para utilizar este tipo de aplicação é necessário ter um computador com alta capacidade de processamento. O uso deste tipo de aplicação em rede conseqüentemente exigirá, além de computadores com alta taxa de processamento, uma rede com alta taxa de transmissão de dados, que suporte o tráfego de informações exigido pelas aplicações, entre os computadores. Assim, diz-se que redes de alta velocidade são aquelas que suportam *altas taxas de transmissão*.

As Redes ATM

A tecnologia de rede ATM (*Asynchronous Transfer Mode*) surgiu da necessidade de ampliação da taxa de transmissão suportada pela tecnologia de redes locais (IEEE 802). Desta maneira, todas as características relativas à tecnologia de rede ATM, foram desenvolvidas tendo-se em vista a viabilização da mais alta taxa de transmissão possível.

Conforme Max Planck (QDS, 1997) "É impossível expressar um princípio realmente novo em termos de um modelo que segue antigas leis". Assim sendo, ATM foi concebida sem que houvesse uma preocupação ou necessidade de conformidade com padrões existentes.

Isto levou a uma nova tecnologia, que diverge muito do padrão Ethernet, o que por si só não torna difícil a integração das redes ATM com os protocolos de redes (LAN) existentes. Sendo, pelo contrário, de fácil integração com eles(MODARRES, 1997).

O benefício mais importante trazido pelas redes ATM – na comparação com as redes locais, é o compartilhamento eficiente do meio. Estabelecida como uma tecnologia de rede ponto-a-ponto, a tecnologia ATM maximiza a largura de banda disponível para transmissão no meio, uma vez que se deixa de compartilhá-lo (MODARRES, 1997), (GUPTA, 1999).

A tecnologia de rede ATM enquadra-se na categoria das tecnologias de redes digitais de serviços integrados (RDSI), ou seja daquelas tecnológicas destinadas para o uso com integração de diferentes tipos de tráfegos (SOARES, 1997).

A tecnologia ATM procura simplificar a arquitetura no nível da camada física de tal modo que qualquer tipo de dado é transmitido de uma mesma maneira. O objetivo principal é diminuir o *overhead* durante o funcionamento da rede, privilegiando a velocidade na transmissão de dados.

Esta abordagem fortalece a opção por uma rede que, além de ser ponto-a-ponto, seja **orientada a conexão**, ou seja, antes que qualquer dado seja transmitido, faz-se necessário o estabelecimento da conexão e a definição do **circuito virtual**.

Em uma rede orientada a conexão, somente após a definição da rota e o estabelecimento da conexão é que o fluxo contínuo de dados inicia-se, diminuindo o *overhead* – durante a transmissão dos dados, e reduzindo o tamanho do cabeçalho de controle do pacote a ser transmitido.

Uma tecnologia de rede ponto-a-ponto que transmite pacotes sobre conexões pré-estabelecidas (orientada a conexão), simplifica o desenvolvimento e o aperfeiçoamento dos comutadores (ou *switch's*) de alta velocidade, porém deixa ao encargo da camada de protocolos de software, todo o controle e a complexidade (ALLES, 1994).

É por isso que ATM tem sido vista como a principal tecnologia para a implantação de redes que suportem aplicações, que incorporem emergentes e sofisticadas características, a saber: processamento e/ou armazenamento distribuído; multimídia; videoconferência; realidade virtual; tele-imersão; etc (TANENBAUM, 1997).

Por suportarem o tráfego de dados, imagem e som em altíssima velocidade e com garantia na qualidade dos serviços, as redes ATM mostram-se atualmente como o estado-da-arte, dentre as tecnologias de redes de computadores.

As características básicas de uma rede ATM – rede ponto-a-ponto e com transmissão orientada a conexão, são predominantemente utilizadas em redes de longa distância. Neste tipo de ambiente, tem-se uma rede de comunicação entre os diversos elementos na rede que alocam e compartilham os enlaces físicos para transmiti-la, mas atualmente, ATM tem despertado interesse para a interconexão de alta velocidade em redes locais (SOARES, 1997).

2.1 A Rede de Multiplexação ATM

Segundo Soares (SOARES, 1997), "A função de comutação (ou chaveamento) em uma rede de comunicação, refere-se à alocação dos recursos da rede (meios de transmissão, repetidores, sistemas intermediários, etc.) para a transmissão pelos diversos dispositivos conectados". Para alocar os recursos da rede sem desperdício, deve-se compartilhá-los.

O compartilhamento de enlaces físicos pode ser feito através de uma técnica chamada **multiplexação**, através da qual pode-se transmitir simultaneamente mais de um sinal num meio físico. Isso é desejável sempre que a banda passante de um enlace físico for maior que a banda passante necessária para a transmissão do sinal. Para multiplexar a transmissão de sinais, existem duas técnicas básicas: a multiplexação por divisão de freqüência e a multiplexação por divisão de tempo.

A técnica de **multiplexação por divisão de freqüência** (ou *Frequency Division Multiplexing* – FDM), baseia-se na divisão da banda passante do enlace, em várias faixas de freqüência menores. Isto é útil quando a largura da faixa de freqüência do enlace é muito maior do que a largura da faixa de freqüência do sinal.

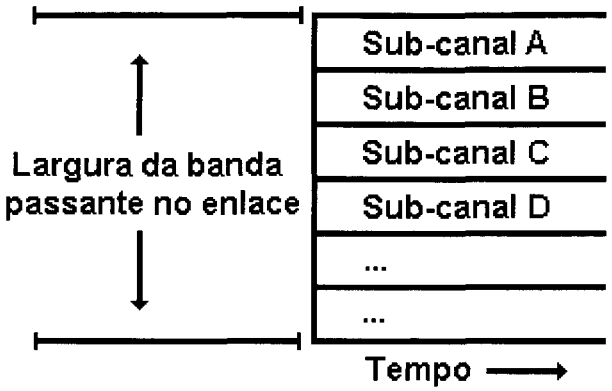

Figura 2: A Técnica de Multiplexação por Freqüência - FDM.
Fonte: (ALVES, 1994). Pág. 142.

Ao dividir a faixa de freqüência do enlace em várias faixas de freqüência menores –uma para cada sinal a ser transmitido, criam-se vários canais de transmissão no mesmo enlace, um para cada sinal a ser transmitido.

Com o uso de uma técnica conhecida como **modulação** é possível o deslocamento (ou *shift*) da freqüência na qual um determinado sinal está sendo transmitido. Quando um sinal é modulado de uma freqüência mais baixa para uma freqüência mais alta, ou vice-versa, o sinal não é alterado, o que possibilita o uso desta técnica.

Assim, antes de serem transmitidos simultaneamente, os sinais são modulados para freqüências disjuntas – dentro da freqüência suportada pelo enlace, de maneira que não colidam. Após a transmissão, os sinais são demodulados o que faz com que as freqüências dos sinais voltem ao normal.

A largura da faixa de freqüência de cada sinal é determinada de acordo com a largura da banda passante do sinal. Desse modo, permite-se que faixas de freqüência em larguras diferentes, sejam transmitidas simultaneamente num enlace.

A **multiplexação por divisão de tempo** (ou *Time Division Multiplexing* – TDM) baseia-se na divisão do tempo de transmissão em pequenas fatias de tempo, o que é útil quando a capacidade de transmissão do enlace é muito maior do que a quantidade de bits gerados pelas estações conectadas ao enlace.

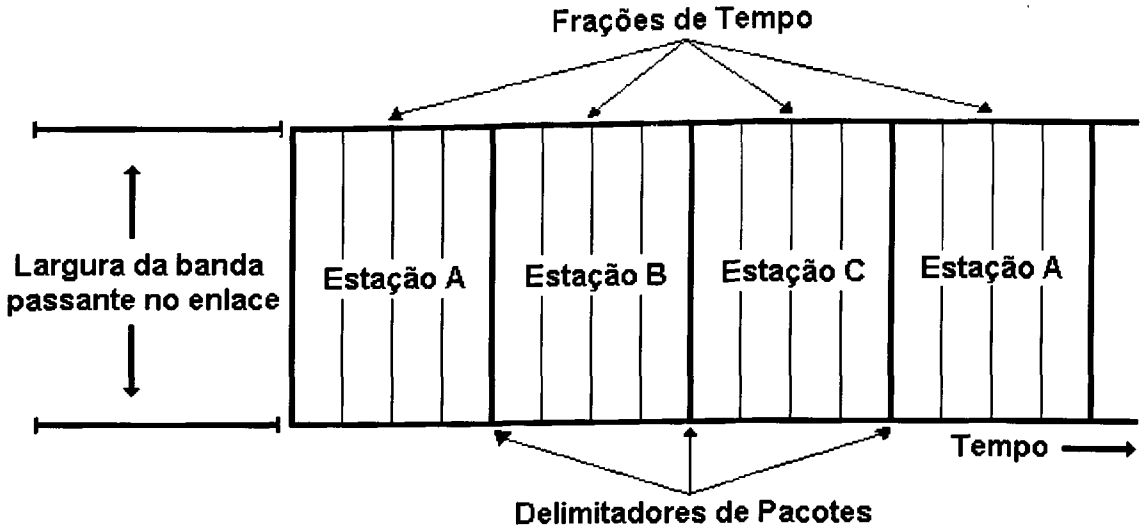

Figura 3: A Técnica de Multiplexação de Tempo - TDM.
Fonte: (ALVES, 1994). Pág. 142.

Através da técnica conhecida como **compartilhamento de tempo** (ou, *time sharing*) divide-se o tempo de transmissão em uma pequena fatia de tempo para cada estação ou canal, que durante sua fatia de tempo usará toda a banda passante do enlace.

A multiplexação por divisão de tempo é influenciada diretamente pelo modo através do qual a transmissão ocorre. Existem dois tipos de multiplexação por divisão de tempo: a multiplexação por divisão síncrona de tempo e a multiplexação por divisão assíncrona de tempo.

A **multiplexação por divisão síncrona de tempo** (ou, *Synchronous Time Division Multiplexing* - STDM) divide em pequenas fatias o tempo de transmissão para cada uma das estações, porém um sinal periódico de sincronismo determina o início e o fim de cada fatia de tempo de uma estação.

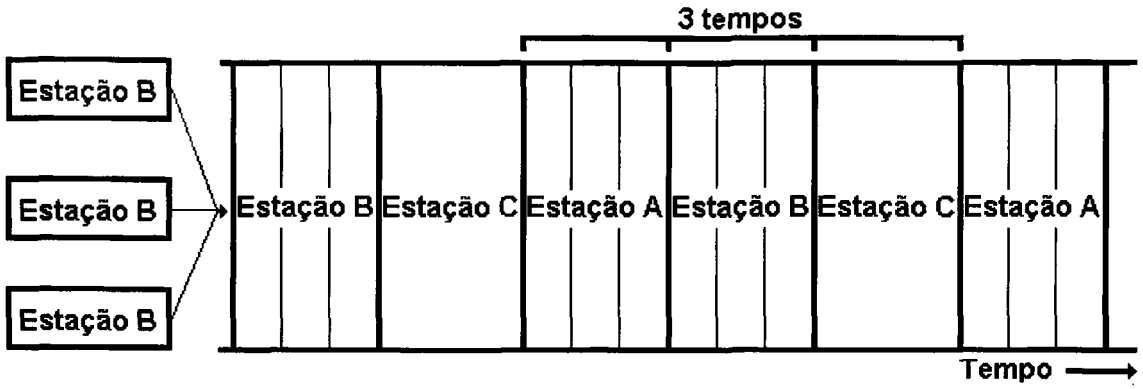

Figura 4: A Técnica de Multiplexação Síncrona de Tempo - STDM.
Fonte: (ALVES, 1994). Pág. 143.

Um inconveniente desta técnica surge quando uma ou mais estações não têm nada para transmitir. Uma vez que uma fatia de tempo lhes é reservada, esta passa a ser então desperdiçada, fazendo com que durante esta fatia de tempo o enlace não seja utilizado.

A **multiplexação por divisão assíncrona de freqüência** (*Asynchronous Time Division Multiplexing* – ATDM) divide em pequenas fatias o tempo de transmissão para cada uma das estações, porém, está baseada numa fila (ou buffer) do multiplexador.

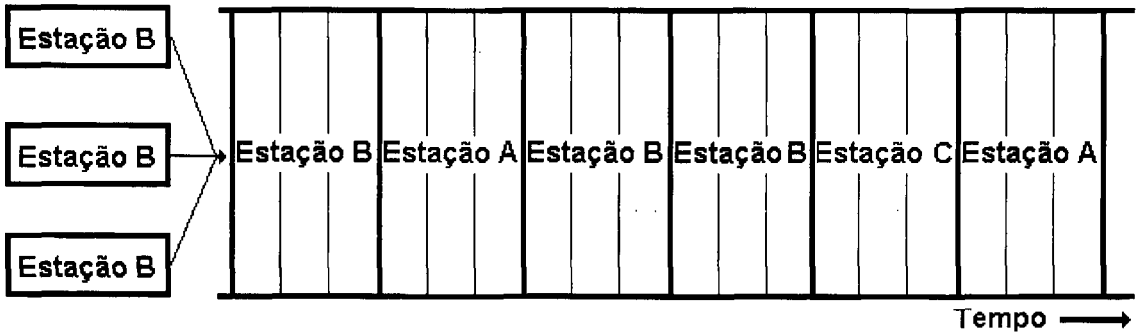

Figura 5: A Técnica de Multiplexação Assíncrona de Tempo - ATDM.
Fonte: (ALVES, 1994). Pág. 144.

Toda vez que uma estação possui dados a transmitir, ela divide os dados em pacotes, que são enviados à fila do multiplexador. Então, a técnica TDM é utilizada sobre os dados que estão na fila para serem transmitidos. Desta maneira, maximiza-se o uso do enlace, uma vez que somente quando nenhuma estação tiver dados a transmitir é que se estará deixando de usar o enlace.

2.2 A Rede de Comutação ATM

Numa rede de comunicação, a comutação pode ser dividida em dois tipos básicos: a comutação de circuitos e a comutação de pacotes. A comutação de pacotes possui variações, nas técnicas de comutação de mensagens e de comutação rápida de pacotes.

A **comutação de circuitos** está baseada na técnica de transmissão orientada a conexão, o que determina que um canal virtual é estabelecido entre a origem e o destino antes que seja efetuada a transmissão de dados. Neste canal virtual, a comunicação entre as duas estações (origem e destino) envolve três fases: o estabelecimento do circuito; a transferência da informação e a desconexão do circuito.

O canal dedicado pode ser: um canal físico (de um ou mais enlaces); um canal físico multiplexado por freqüência (de um ou mais enlaces) ou um canal físico multiplexado por tempo (de um ou mais enlaces).

A comutação por circuito é mais adequada para a transmissão de mensagens nas situações em que o fluxo de informações é constante – o que gera um tráfego de dados com uma taxa de bits constante. Nessa situação, após o estabelecimento da conexão, o **canal virtual** torna-se dedicado, minimizando o *overhead* causado pelo empacotamento, pelo endereçamento, e pelo roteamento que a comutação por pacotes possui.

Existe, neste caso, a garantia da taxa de transmissão, do momento em que a conexão é estabelecida até que a desconexão seja efetuada, o que determina uma melhor QoS (*Quality of Service*). Esta garantia pode, por outro lado, gerar ociosidade, caso o tráfego não tenha taxas de bits constantes, além de poder acarretar a não aceitação de novas conexões devido à falta de recursos (largura de banda no enlace).

Para transmissões com um tráfego de dados que se caracterize por uma taxa de bits variável, ou tráfego em rajadas (ou, *burst*), a comutação de circuitos não é

recomendada, porque os canais são alocados com base na taxa de pico, o que os torna ociosos durante qualquer taxa menor de transmissão.

A **comutação de pacotes** está baseada na técnica "armazena-e-encaminha" (ou, *store-and-forward*) que não é orientada a conexão. Nela, as mensagens a serem transmitidas carregam o endereço do destino.

Um pacote enviado por uma estação através de um determinado enlace é retransmitido, enlace a enlace, até chegar ao destino. Cada nó recebe o pacote, analisa o seu endereço de destino e encaminha-o para a rota devida.

A **comutação de mensagens** é uma técnica semelhante que, assim como a comutação de pacotes, está baseada na técnica "armazena-e-encaminha". O que diferencia essas técnicas é que, numa rede de comutação de pacotes, os pacotes possuem tamanho fixo e numa rede de comutação de mensagens, as mensagens não possuem tamanho pré-determinado. Conseqüentemente, os comutadores nas redes de comutação de pacotes devem suportar mensagens relativamente grandes, o que os encarece.

Tanto a comutação de pacotes, quanto à comutação de mensagens, são apropriadas para os tráfegos com taxa de bits variável, normalmente gerados por aplicações com: som, vídeo comprimido, texto e imagens, definidas anteriormente como aplicações sofisticadas.

Nestas técnicas, cada pacote (ou mensagem) somente utiliza o enlace durante a transmissão, sendo a alocação do meio feita "dinamicamente" para cada pacote (ou mensagem) em cada um dos enlaces durante toda a rota, até o destino.

Mesmo que o tráfego nos enlaces seja alto, os pacotes (ou mensagens) são aceitos pelos nós, porém, o congestionamento das linhas e o aumento da fila de transmissão acarretam um maior tempo para a transferência.

Pode-se citar como vantagem de usar a comutação de pacotes, em relação à comutação de mensagens, os seguintes fatos: ela permite enviar várias partes de uma mensagem simultaneamente e possui menores requisitos para os nós comutadores (principalmente no que se refere ao armazenamento).

Um outro aperfeiçoamento da comutação de pacotes é a chamada **comutação rápida de pacotes**, na qual os serviços executados pelos nós comutadores, na camada física e de enlace, são otimizados ou dispensados. Com sua natureza orientada a conexão, a tecnologia ATM dispensa os serviços da camada de rede, pois as três camadas baixas são otimizadas, diminuindo ao máximo o desperdício de tempo em controle.

Os serviços de reconhecimento e retransmissão não são feitos no nível de enlace, ficando para controle fim-a-fim. Desse modo, canal que transmite os pacotes não conduz informações adicionais de sinalização, para as quais existem canais específicos.

Nas redes ATM, os pacotes transmitidos são chamados **células**. O tamanho pequeno e fixo dessas células, diminui a complexidade dos comutadores, além de minimizar o tempo de empacotamento (SOARES, 1997).

Assim, os nós processam menos informações, alcançando uma maior velocidade de armazenamento e encaminhamento, tendo como conseqüência uma taxa maior de transferência de informações.

Segundo Soares: "As técnicas de comutação rápida de pacotes podem ser baseadas na transferência de unidades de informação de tamanho variável (ou, *frame relay*) ou de tamanho fixo (ou, *cell relay*)".

Estas características tornam as redes de transferência de células a base para as redes de serviços integrados, pois a simplificação de sua arquitetura do nível da camada

física até a camada de rede permite transmitir qualquer tipo de dado de uma mesma maneira.

2.3 A Rede Digital de Serviços Integrados ATM

As Redes Digitais de Serviços Integrados – RDSI (ou, *Integrated Services Digital Network - ISDN*) surgiram como uma tecnologia de transmissão de dados que suporta uma grande quantidade de serviços distintos, em um ambiente onde o tráfego tenha características variadas.

O termo RDSI foi definido pela ITU (*International Telecommunications Union*) quando esta ainda chamava-se CCITT, e a primeira definição na recomendação G.702 definia ATM como: "Uma rede digital integrada, na qual os mesmos comutadores e caminhos digitais são usados para os diferentes serviços, por exemplo: telefonia e dados".

Para melhor estabelecer o suporte a diferentes serviços, o ITU classificou os serviços em quatro categorias: Serviços Conversacionais, Serviços de Recuperação, Serviços de Mensagem e Serviços de Distribuição.

- Os **Serviços Conversacionais** fornecem meios para transferências fim-a-fim em tempo real. Algum desses serviços, estão associados, por exemplo, às aplicações em vídeo-conferência, vídeo-telefonia, etc.
- Os **Serviços de Recuperação**, fornecem a facilidade para a recuperação de informações armazenadas remotamente serviços dessa natureza são usados, por exemplo, nas aplicações do tipo: vídeo sob demanda, livraria eletrônica, biblioteca virtual, museu virtual, etc.
- Os **Serviços de Mensagem** permitem a comunicação entre usuários via unidades de armazenamento (armazena-e-encaminha), nas aplicações que não necessitam de tempo real como: correio de vídeo, correio de documentos multimídia, etc.

- Os **Serviços de Distribuição,** por sua vez, são utilizados em aplicações que necessitem distribuir dados, tais como: distribuição de áudio e vídeo, distribuição de documentos, difusão de TV, etc., podendo este serviço ser feito com ou sem o controle do usuário.

Considerando os tipos de serviço que suportam, as Redes Digitais de Serviços Integrados dividem-se em: RDSI de Faixa Estreita (RDSI-FE ou, *Narrowband Integrated Services Digital Networks* – N-ISDN) e RDSI de Faixa Larga (RDSI-FL ou, *Broadband Integrated Services Digital Networks* – B-ISDN).

2.3.1 A Rede Digital de Serviços Integrados de Faixa Estreita

A RDSI-FE surge com a proposta mais viável atualmente, para o provimento de acesso digital aos assinantes da rede de telefonia, sem que toda a infra-estrutura analógica existente tenha que ser substituída, uma vez que a proposta de uma RDSI-FE única, obviamente não pode ser obtida em larga escala, mesmo que em médio prazo, dada a necessidade de digitalização de toda a infra-estrutura de rede telefônica existente.

Ressalte-se que apesar de não estar integrada diretamente à infra-estrutura que suporta os aparelhos analógicos existentes, a tecnologia digital de faixa estreita oferece **serviços integrados** de voz, vídeo, dados, etc. numa mesma linha digital, podendo ser usada portanto, com a integração total de serviços.

A instalação de uma rede única integrada exige uma profunda modificação no sistema de convicção existente, desde a rede de comutação de pacotes (infra-estrutura da rede) até os aparelhos utilizados pelos usuários. O que torna a mudança mais difícil, mas, factível em longo prazo.

As RDSI-FE são caracterizadas por trabalhar com o **modo de transferência síncrono** (ou, *Synchronous Transfer Mode* – STM), e com modulação por divisão de tempo TDM, suportando taxas de transmissão de até 2.048 Mbps.

O acesso digital pode ser feito através de **pacotes** de canais, que formam a **estrutura de acesso**. A combinação destes canais pode ser feita de acordo com os seguintes tipos padronizados de canais:

- canal A: 4 kHz – canal de telefone analógico;
- canal B: 64 kbps – canal digital PCM (*Pulse Code Modulation*) para voz e dados;
- canal C: 8 ou 16 kbps – canal digital;
- canal D: 16 ou 64 kbps – canal digital para sinalização externa (*out-of-band*);
- canal E: 64 kbps – canal digital para sinalização interna (ISDN); e
- canal H: 384-, 1536-, ou 1920 kbps – canal digital.

A recomendação I.412 da ITU-T define as seguintes combinações possíveis para a interface do assinante: a Estrutura de Acesso Básico (ou, Ligação Básica - LB), a Estrutura de Acesso Primário (ou, Ligação Multiplexada Primária - LMP) e a Estrutura de Acesso Híbrida. Ver Figura 6.

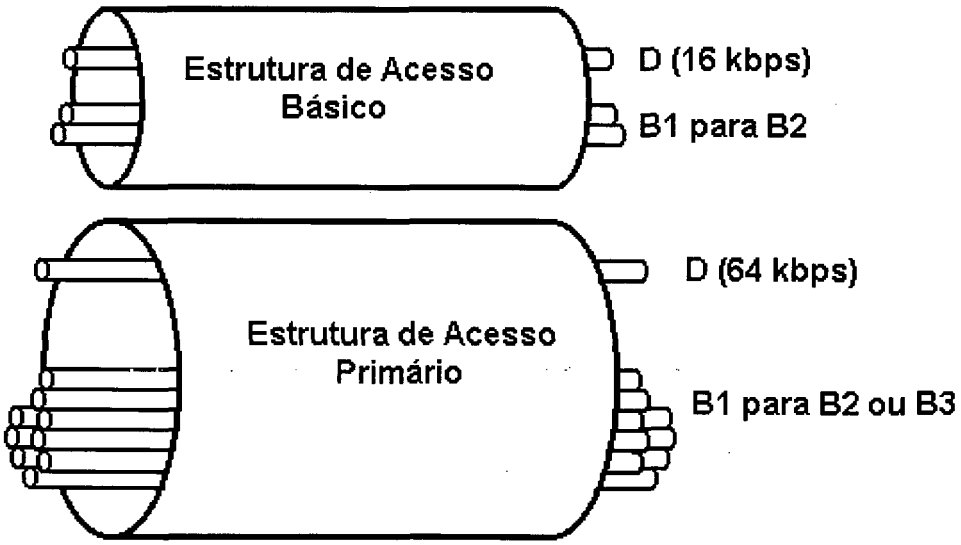

Figura 6: Estruturas de acesso.
Fonte: (TANENBAUM, 1997). Pág. 143.

A **estrutura de acesso básico** (que se destina a usuários e/ou assinantes com um uso menor) foi padronizada como 2B+D, ou seja, dois Canais B de 64 Kbps mais um Canal D de 16 Kbps de sinalização. Os Canais B de 64 Kbps são independentes, podendo transmitir informações distintas (e.g. 64 Kbps para fax e 64 Kbps para dados) ou não (e.g. 128 Kbps para dados).

Somando-se a estes 128 Kbps os 16 Kbps do canal de sinalização chega-se a taxa de transmissão de 144 Kbps. Pulsos adicionais de sincronização e de delimitação de quadros podem elevar a taxa de transmissão para 192 Kbps (WIRTH, 1994), (TANENBAUM, 1997).

A **estrutura de acesso primário** (destinada ao usuário e/ou assinantes que necessitam de maiores taxas de transmissão) foi padronizada em dois tipos de acessos: o tipo T1 (23B+D) com vinte e três Canais B de 64 Kbps e um Canal D de sinalização (de 64 Kbps), atingindo 1,544 Mbps, e o tipo E1 (30B+D) com trinta Canais B de 64 Kbps e um Canal D de sinalização (de 64 Kbps), atingindo 2,048 Mbps.

O uso e a implantação de uma RDSI-FE é um primeiro passo no caminho para uma completa Rede Digital de Serviços Integrados de Faixa Larga, porque aproveita a atual rede de telefonia com pares de cobre. Ver Figura 7.

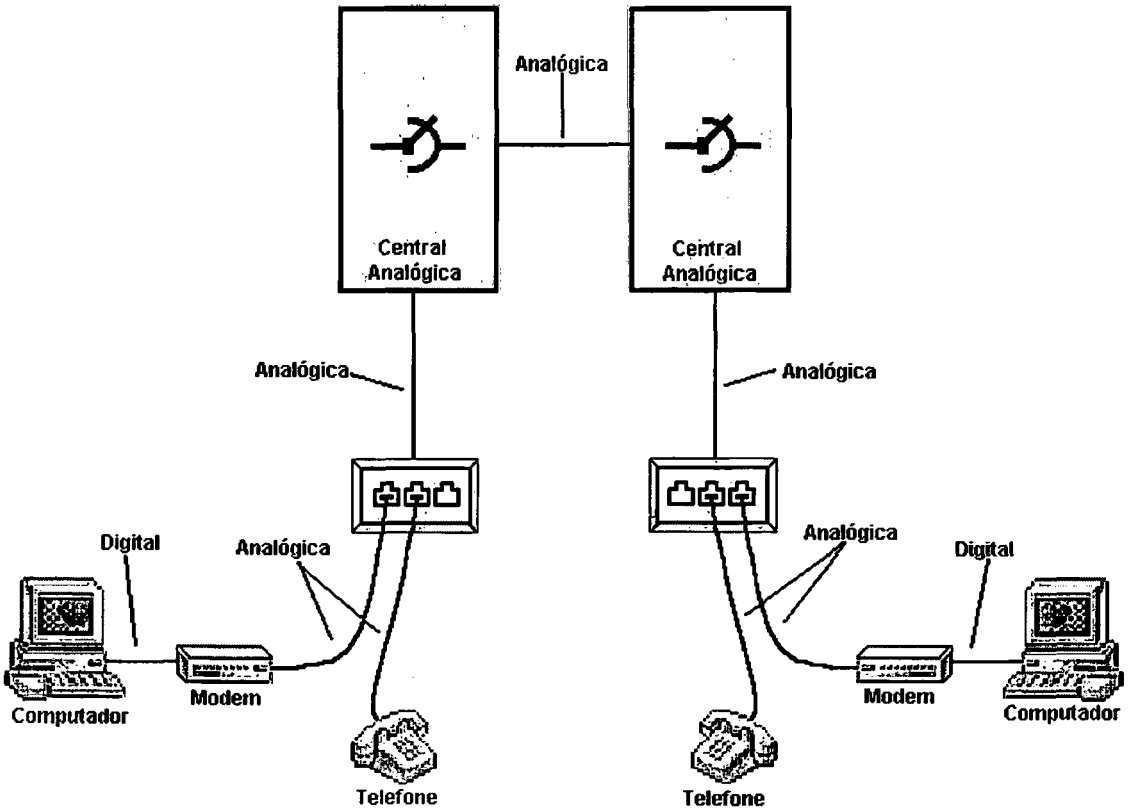

Figura 7: A rede de telefonia analógica.
Fonte: (WIRTH, 1994). Pág. 3.

Com o aumento crescente da demanda por taxas de transmissão aos poucos a rede de telefonia passou a ser atualizada com a implantação de enlaces digitais entre as centrais e também com a digitalização das centrais telefônicas. Portanto, onde esta atualização ocorre, passa-se a ter uma rede de telefonia com uma parte analógica e com uma parte digital (tal como em grande parte do Brasil hoje). Ver Figura 8.

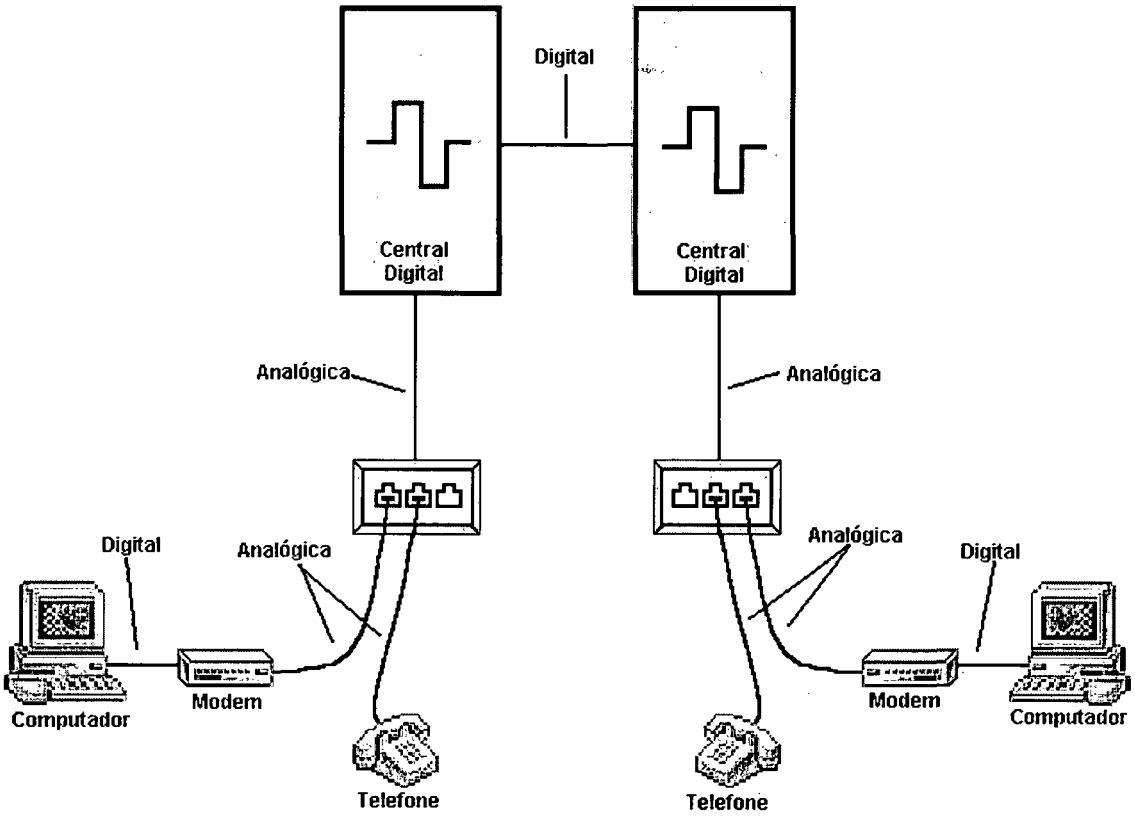

Figura 8: A rede de telefonia mista.
Fonte: (WIRTH, 1994). Pág. 4.

O objetivo, a longo prazo, é criar uma rede digital que não sirva só para a telefonia, mas também para outros serviços, formando assim uma rede de fato digital (ou seja, digital fim-a-fim) e com serviços integrados. Onde a digitalização da rede de telefonia já ocorreu, faz-se necessário levar o sinal digital à casa (ou à empresa) do assinante. Para fazer isto, há que se considerar a forma pela qual o assinante está conectado à rede de telefonia.

Os pares de cobre, que conectam o assinante à rede de telefonia, suportam somente serviços de faixa estreita, que limitam o assinante a 2.048 Kbps, em um acesso multiplexado primário 30B+D. Ver Figura 9.

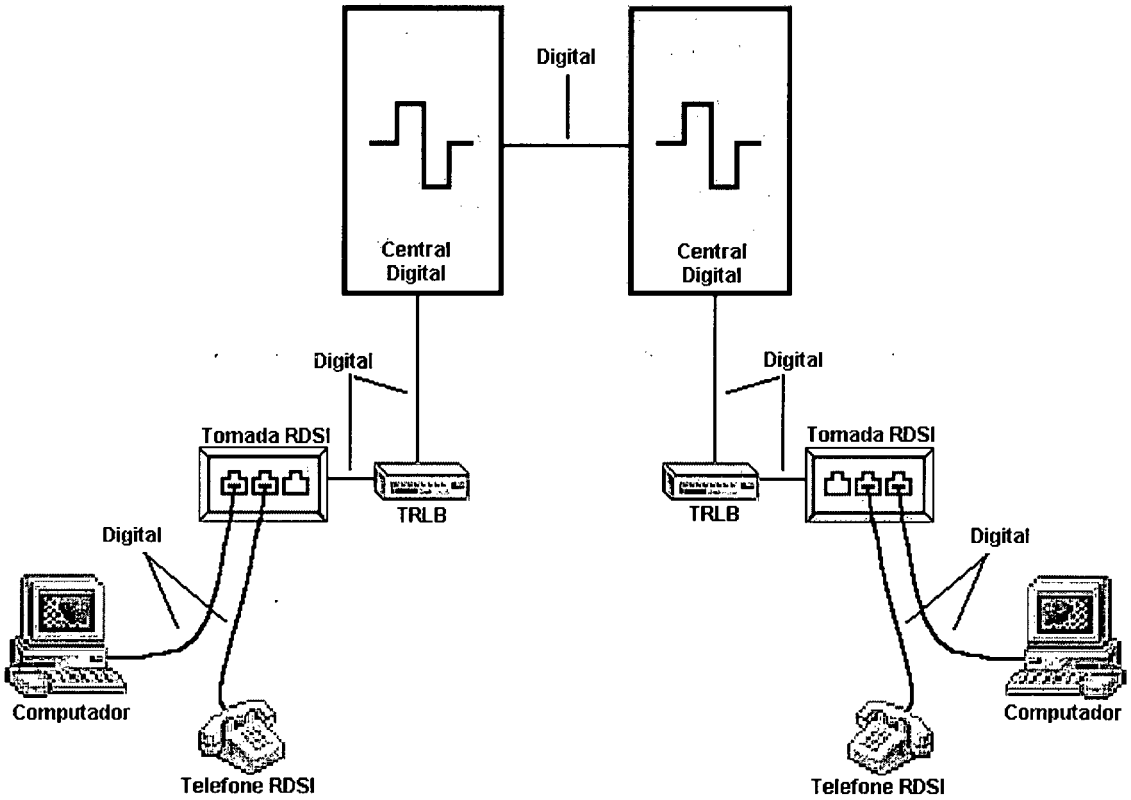

Figura 9: A rede de telefonia digital de serviços integrados.
Fonte: (WIRTH, 1994). Pág. 4.

Como visto anteriormente, com o tempo os tradicionais pares de cobre serão substituídos por outros meios de transmissão capazes de suportar taxas mais altas de transmissão, ultrapassando a capacidade dos Canais B e D. O primeiro passo rumo a esta infra-estrutura digital que suporte RDSI-FL é usar a tecnologia de RDSI-FE, com uma estrutura mista com centrais digitais e acesso de assinantes feito sobre os pares de cobre analógicos (já instalados).

2.3.2 A Rede Digital de Serviços Integrados de Faixa Larga

O que caracteriza uma aplicação (ou um tráfego) como sendo de faixa larga, é, segundo Soares (TANENBAUM, 1997) "O fato de lidarem com objetos não convencionais (áudio, vídeo, etc.), de exigirem transferência contínua de dados a altas taxas e além de tudo, exigirem acesso sincronizado aos dados".

O ITU define "serviços de faixa larga" como sendo serviços que demandam canais para suportar taxas de transmissão maiores do que aquelas do acesso primário (ou, LMP) das redes de faixa estreita (até 2 Mbps). Em sua recomendação I.211, o ITU determina os requisitos básicos como sendo: requisitos de multimídia, qualidade de serviço (QoS), temporização, sincronização e sinalização.

Uma nova infra-estrutura de rede digital (sem utilizar os pares de cobre), deverá suportar **serviços de faixa larga** (com taxas de transmissão maiores do que dois Mbps – em Canais do Tipo H). Ver Tabela 01.

Tabela 01: Os canais do Tipo H

Canais	Hierarquia de 2 Mbps	Hierarquia de 1,5 Mbps
H0	384 Kbps	
H1	1.920 Kbps	1.536 Kbps
H2	32.768 Kbps	Entre 43.000 e 45.000 Kbps
H3	Entre 132.032 Kbps e 138.240 Kbps	

Para que fosse possível transmitir nas altas taxas exigidas de uma RDSI-FL, pensava-se, inicialmente, aumentar as estruturas de acesso das RDSI-FE, ampliando o suporte para um número maior de canais multiplexados num único acesso, tal que houvesse uma fácil compatibilização entre as RDSI.

Esta proposta de adequação da tecnologia STM para dar suporte a RDSI-FL, logo foi recusada devido à alta complexidade necessária para gerenciar estruturas com muitos canais. O número de canais necessários para transmitir em altas taxas (100 Mbps, 150 Mbps, 622 Mbps, etc.) seria muito grande, dificultando esta adaptação.

Desta maneira, começou-se a questionar a capacidade e a flexibilidade do modo síncrono de transmissão STM, para trabalhar com as estruturas de acesso suportadas pela RDSI-FL. Assim, o modo assíncrono de transferência começou a ser cogitado para uma nova faixa de transmissão que permitisse a transmissão de dados em redes de faixa larga.

O **modo assíncrono de transferência** (*Asynchronous Transfer Mode* - ATM), teve a sua primeira recomendação aprovada pelo ITU para as Redes Digitais de Serviços integrados de Faixa Larga – RDSI-FL (ou, B-ISDN), em 1988 (I.121).

A tecnologia ATM transpõe as limitações do modo de transferência síncrono (ou, STM), de diversas maneiras. Por sua natureza não sincronizada, as redes ATM suportam melhor um tráfego de fluxo variável, o que é predominante em um ambiente de **serviços integrados** – que são, supostamente, heterogêneos, não deixando a desejar também no desempenho com tráfegos de fluxo constante.

Em 1991, como resultado de um consórcio de empresas de informática e telecomunicações, e organizações interessadas na aceleração do desenvolvimento e da implantação da tecnologia ATM, surgiu o **ATM Forum**. O ATM Fórum vem sendo responsável por uma série de recomendações e acordos, e apesar de não ter nenhuma autoridade para definir padrões internacionais oficiais, vem contribuindo para o desenvolvimento da tecnologia.

"As Redes baseadas em ATM podem ser projetadas de forma a aproveitar melhor os meios de comunicação na presença de tráfego em rajadas e ao mesmo tempo, garantir retardo máximo para serviços que suportam fontes de tráfego contínuo". (SOARES, 1997) p. 534.

2.4 O modelo de referência para protocolos da RDSI-FL

As redes ATM são organizadas (assim como as redes de longa distância tradicionais), com base nos enlaces e nos *switch's* (com função de roteadores), baseando-se na transmissão de pequenas unidades de informação, de tamanho fixo, chamado **células**.

O desenvolvimento da tecnologia de rede ATM foi feito com base num modelo de referência próprio, diferente do modelo de referência OSI e do modelo TCP/IP

(TANENBAUM, 1997). O **Modelo de Referência para Protocolos da RDSI-FL** (*Protocol Reference Model* - PRM), definido pela ITU, consiste de três planos (o plano de usuário, o plano de controle e o plano de gerenciamento), além de três camadas inferiores mais uma camada superior. Ver Figura 10.

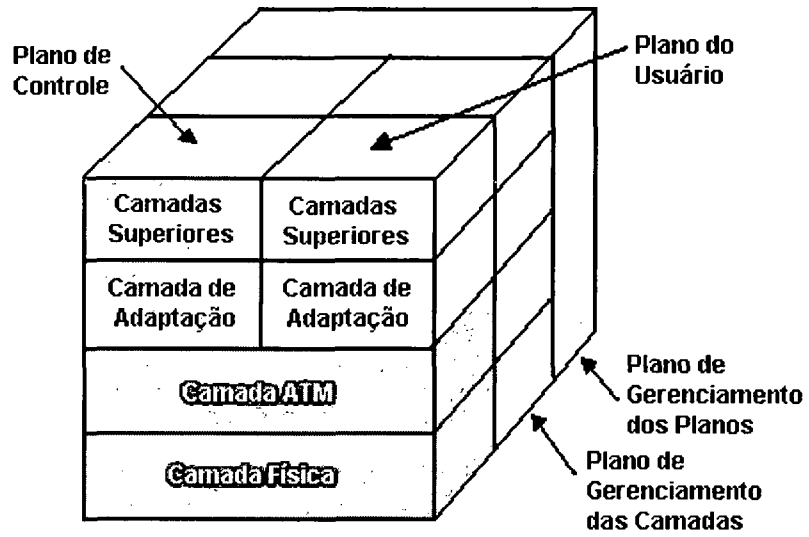

Figura 10: O modelo de referência de protocolos da RDSI-FL.
Fonte: (SOARES, 1997). Pág. 547.

O **Plano de Usuário** é utilizado para a transferência de informações do usuário. O fluxo de informações que ocorre entre as camadas (física, ATM, adaptação e as camadas superiores) se dá neste plano, assim como o controle do fluxo através de monitoração e correção de erros.

Utiliza-se o **Plano de Controle** para funções de controle das chamadas e das conexões. Considerando a natureza orientada a conexão das redes ATM, compreende-se a necessidade de um plano responsável por estabelecer, monitorar e finalizar conexões.

O **Plano de Gerenciamento** é responsável pelo gerenciamento da rede como um todo, tendo suas funções divididas em funções básicas, a saber: o gerenciamento de planos e de camadas. A gerência de camadas, também dividida em camadas, monitora o fluxo de informações em cada camada. A gerência de planos gerencia os planos de usuário, de controle e de gerência.

A **Camada Física** é responsável: pela conversão eletro-óptica, pela sinalização na linha, pelo alinhamento e transmissão dos bits, ou seja, pelo meio físico em si, além de controlar os erros nos cabeçalhos das células e gerenciar o fluxo de células, inserindo células especiais de preenchimento, quando for necessário, com a intenção de que seja mantido um fluxo de células constante. A Camada Física é dividida em duas subcamadas: a subcamada de Meio Físico – PM (*Physical Medium*) e a subcamada de Convergência de Transmissão – TC (*Transmission Convergence*).

A **Camada ATM** é responsável pela multiplexação e de-multiplexação de células, pela adição e remoção de cabeçalho de células, pelo chaveamento e encaminhamento de células com base no cabeçalho, e também pelo controle genérico de fluxo (*Generic Flow Control* – GFC). Não é subdividida em outras subcamadas.

A **Camada de Adaptação** tem como funções: a quebra e a remontagem do fluxo de informações em fragmentos a serem transmitidos, célula a célula; a multiplexação de serviços; a detecção de perdas de células; e a recuperação da ordem das informações. Essa camada é subdividida em duas subcamadas: a subcamada de Quebra e Remontagem – SAR (*Segmentation and Reassembly*) e a subcamada de Convergência – CS (*Convergence Sub-layer*). Ver Figura 11.

CAMADAS	SUB-CAMADAS	FUNÇÕES
AAL	CS	Convergencia
	SAR	Quebra e Remontagem
ATM		Controle genérico de fluxo Inserção e remoção de Cabeçalho Interpretação de VPI/VCI Multiplicação/Demultiplicação de células
PHY	TC	Desacoplamento de taxa de células Geração e verificação de HEC Delineamento de células Geração e Recuperação de Frames
	PM	Transmissão pelo meio físico Conversão eletro-ótico

Figura 11: As camadas no modelo de referência de protocolos ATM.
Fonte: (SOARES, 1997). Pág. 548.

2.5 A Camada Física

A **camada física** está implementada em todo equipamento de rede, e é dividida em duas subcamadas: a subcamada de Meio Físico – PM (*Physical Medium*) e a subcamada de Convergência de Transmissão – TC (*Transmission Convergence*). A **Subcamada de Meio Físico** recebe da Subcamada TC os dados a transmitir e é responsável pela adequação dos dados para que, bit a bit, sejam transmitidos através do meio físico. Esta tarefa pode ser detalhada em algumas outras: a conversão eletro-óptica dos bits, a sinalização na linha de comunicação, o alinhamento dos bits, e a transmissão dos bits através do meio físico.

A **Subcamada de Convergência de Transmissão** ocupa-se com três tarefas básicas: o controle dos erros na transmissão das células, a gerência do fluxo de células, o delineamento de células, e o embaralhamento.

A verificação de erros do cabeçalho (*Header Error Check* – HEC) é feita através de uma informação de redundância para a detecção de erros, que é armazenada em um campo (o HEC) no cabeçalho da célula ATM. Cabe à Subcamada TC o cálculo e a inserção/verificação do HEC, no cabeçalho da célula.

O "controle do fluxo de células" trata do desacoplamento da taxa de transmissão em relação à taxa de geração de células. Isto é feito através da inserção de células especiais de preenchimento, quando necessário, para que o fluxo de células seja constante, mantendo-se compatível com a taxa de transmissão utilizada no meio.

O "delineamento das células" é a preparação do fluxo de bits (ou bytes), através da inserção de bits (ou bytes) que determinam onde começa e termina uma célula.

O "embaralhamento" (ou *scrambling*) de parte dos bits de uma célula pode ser feito para evitar longas seqüências de **uns** ou **zeros**, evitando também que os dados da célula possam ser confundidos com o cabeçalho.

2.6 A Camada ATM

A camada ATM, a exemplo da Camada Física, encontra-se implementada em qualquer elemento ativo de rede, não sendo porém, subdividida em outras subcamadas. O processamento efetuado na Camada ATM ocorre sobre o cabeçalho das células.

O ITU decidiu por uma célula pequena com 53 bytes (ou octetos), sendo 48 de dados e 5 de cabeçalho – ver Figura 12, optando pela rapidez de manipulação (empacotamento e desempacotamento) e simplificando o desenvolvimento de hardware. O único argumento contra o uso de uma célula pequena é o *overhead* adicional do cabeçalho em relação aos dados transmitidos nas células.

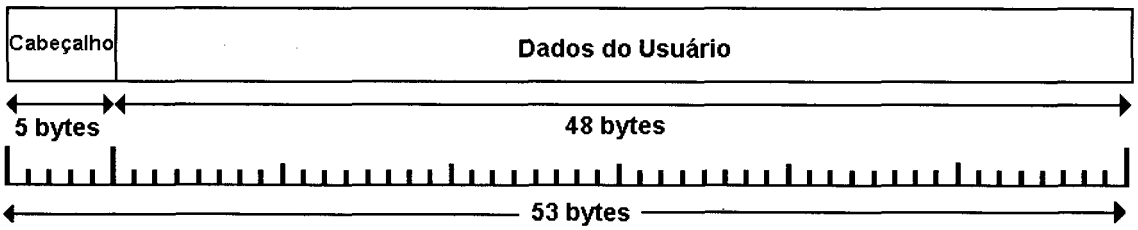

Figura 12: A célula ATM.

No nível da camada ATM, existem duas interfaces distintas: a Interface Usuário-Rede (*User-Network Interface* – UNI) e a Interface Rede-Rede (*Network-Network Interface* – NNI). Cada interface possui um cabeçalho específico, e em ambos os casos o tamanho é de 40 bits, ver Figura 13.

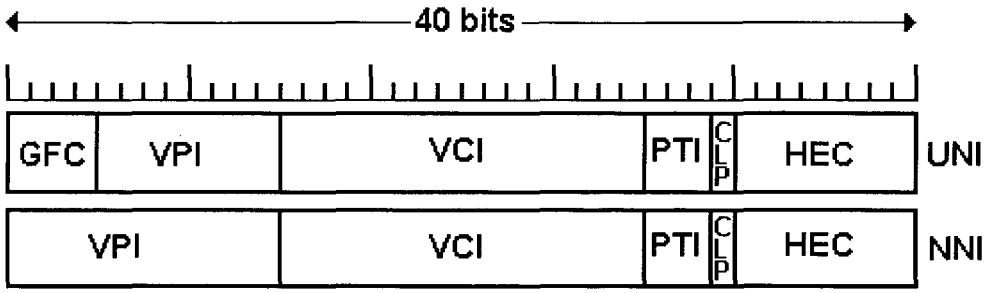

Figura 13: Os cabeçalhos de células ATM.
Fonte: (TANENBAUM, 1997). Pág. 451.

Os campos VPI (*Virtual Path Identifier*) e VCI (*Virtual Channel Identifier*) são necessários para viabilizar a comutação das células. O campo PTI (*Payload Type Identifier*) determina o tipo da célula (indicando se é uma célula de dados de usuário ou uma célula de informação de manutenção de comutadores ou, ainda, uma célula de gerência de recursos), ver Tabela 02.

Tabela 02: Os valores do campo PTI.

PTI	Interpretação
000	Célula de informação de usuário, não passou por caminho congestionado
001	Célula de informação de usuário, passou por caminho congestionado
010	Célula de informação de usuário, não passou por nó congestionado
011	Célula de informação de usuário, passou por nó congestionado
100	Informação de manutenção entre comutadores adjacentes
101	Informação de manutenção entre comutadores origem e destino
110	Célula de gerenciamento de recursos
111	Reservado para uso futuro

Fonte: (SOARES, 1997). Pág. 572.

O campo CLP (*Cell Loss Priority*) indica a prioridade da célula para uma eventual necessidade de descarte de células. As células cujo campo CLP possui valor igual a "1" são descartadas antes das células cujo campo CLP possui valor igual a "0". O campo HEC (*Header Error Check*) – conforme detalhado anteriormente, corresponde a um código de verificação de erros do cabeçalho. O campo GFC (*General Flow Control*) aparece somente na Interface Usuário-Rede (UNI)

As células geradas para fazer o desacoplamento da taxa de células da rede não seguem o formato padrão de células aqui descrito. Elas estão restritas à Camada Física,

não sendo, portanto, passadas à Camada ATM, e seu formato aparece representado na Figura 14.

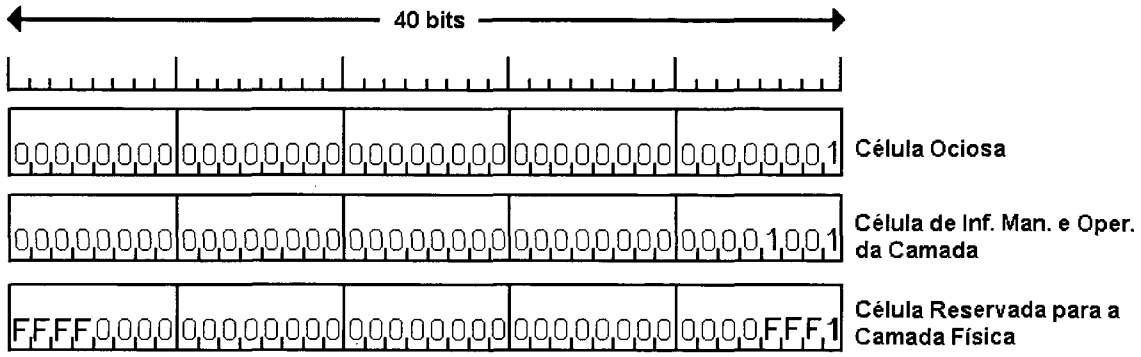

Figura 14: O formato do cabeçalho das células especiais.

A célula é o elemento básico de uma rede ATM. Todas as células são transportadas por uma conexão fim-a-fim. As conexões fim-a-fim em redes ATM são chamadas de **conexão com canal virtual** (ou *Virtual Channel Connection* – VCC).

Um **enlace de canal virtual** (ou, *Virtual Channel Link* – VCL) é definido como sendo cada um dos enlaces estabelecidos entre os comutadores que fazem parte de um determinado VCC. Ver Figura 15.

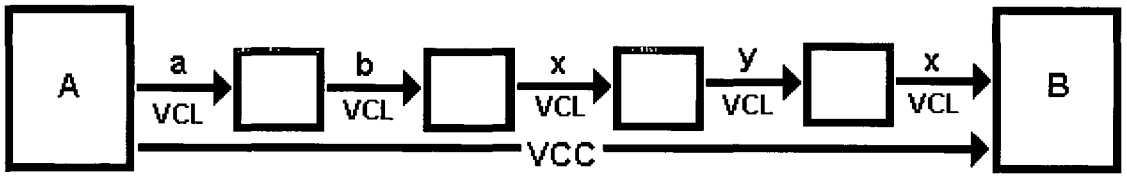

Figura 15: A conexão com canal virtual e os enlaces de canal virtual
Fonte: (SOARES, 1997). Pág. 575.

Conforme ilustrado na Figura 15, após o estabelecimento de uma conexão (ou melhor, de uma conexão com canal virtual - VCC), tem-se a série de VCL's que determinam, comutador a comutador, ou seja, enlace a enlace, o caminho que vai da origem até o destino.

Segundo Soares (SOARES, 1997), "Uma conexão com canal virtual é formada pela concatenação de conexões virtuais estabelecidas nos vários enlaces da rede, da

origem até o destino, formando um caminho único através do qual as células são encaminhadas". Observa-se ainda que um mesmo enlace entre dois comutadores pode suportar vários VCL's de VCC's distintos.

Cada comutador tem várias portas de entrada e de saída associadas aos enlaces físicos nele conectados. Sua função é receber células pelas portas de entrada (por um determinado VCL) e transmiti-las adiante pela respectiva porta de saída (outra VCL). O comutador tem como estrutura básica os seguintes componentes: Controladores de Entrada (ou CE's); Controladores de Saída (ou CS's); Elemento Comutador; e Processador de Controle.

Todo o processo de comutação é efetuado em nível de hardware com a participação dos CE's, CS's e do elemento comutador. O processador de controle é utilizado para: o estabelecimento e o rompimento das conexões; a manutenção das tabelas de VCI e VPI; além de manutenção e gerenciamento. Ver Figura 16.

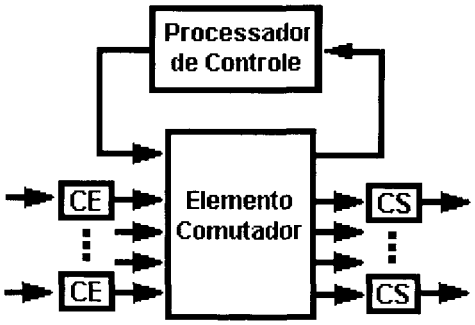

Figura 16: A estrutura de um comutador ATM.
Fonte: (SOARES, 1997). Pág. 581.

As informações relativas à origem e ao destino das células são determinadas no momento do estabelecimento da conexão e ficam armazenadas em uma **tabela de rotas**, no comutador.

Cada célula traz, em seu cabeçalho, a informação que determina qual é o VCL no qual ela está trafegando. Ao receber uma célula, por uma determinada porta de entrada,

o comutador utiliza a tabela de rotas para determinar a porta de saída e a VCL de destino. Para cada porta de entrada, há uma tabela de rotas (ou de comutação).

No cabeçalho das células ATM estão as informações que determinam o tipo da célula (PTI), a prioridade da célula (CLP), e a verificação de erro (HEC). Além disso, dois campos determinam a rota da célula: VCI e VPI. Os campos VPI e VCI identificam o caminho e o enlace (dentro do caminho), através dos quais, o comutador está conectado ao comutador seguinte. Uma vez que os caminhos concentram em si um ou mais enlaces, o conjunto VPI/VCI está vinculado no comutador a uma determinada porta de saída. Ver Figura 17.

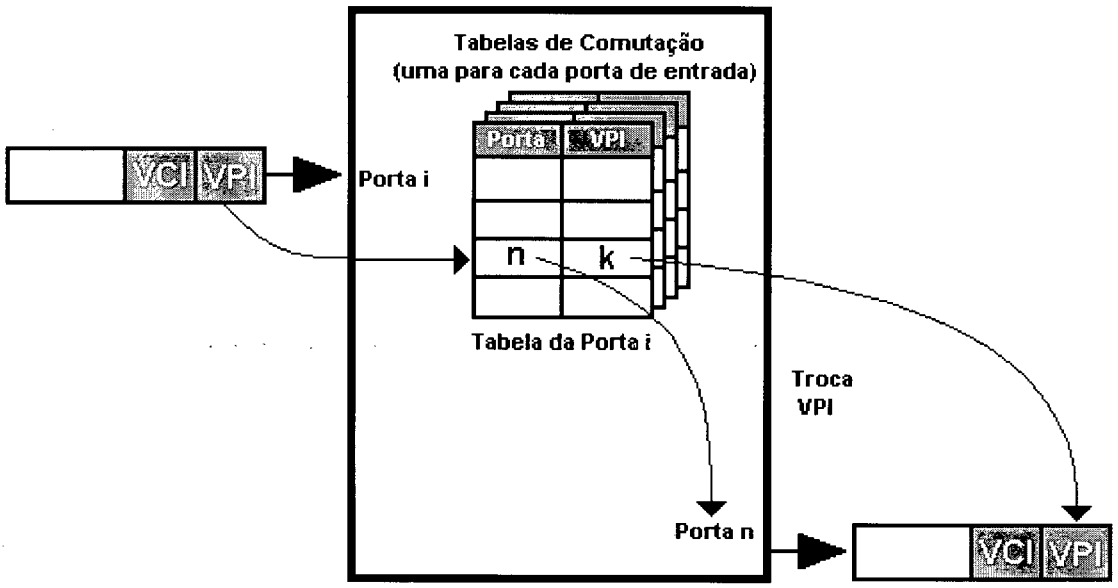

Figura 17: O endereçamento da célula (VPI/VCI).
Fonte: (SOARES, 1997). Pág. 577.

Ao associar, nas tabelas de rotas de cada porta de entrada dos comutadores, todas as VCC's estabelecidas, criam-se enormes tabelas de rotas a serem processadas. Para reduzir esta quantidade de informações e objetivando, principalmente, diminuir o *overhead* que isto provoca, é possível reunir vários VCC's roteados pelos mesmos caminhos, pelo menos por alguma parte da rota.

O grupo de VCC's comutadas em um mesmo caminho é chamado de **conexão de caminho virtual** (VPC, ou *Virtual Path Connection*). Analogamente à conexão de canal

virtual, uma conexão de caminho virtual é formada pela concatenação de vários enlaces, estes por sua vez são chamados de **enlace de caminho virtual** (VPL, ou *Virtual Path Link*).

Em um meio físico de transmissão, que interconecta dois comutadores, podem-se ter várias conexões virtuais. Estas conexões são, em primeira instância, caminhos virtuais VPL's, reconhecidos pelos comutadores através de identificadores de caminhos virtuais VPI. Os VCL's, por sua vez, podem reunir vários enlaces de canais virtuais VCL's, que são reconhecidos pelos comutadores através de identificadores de canais virtuais VCI. Ver Figura 18.

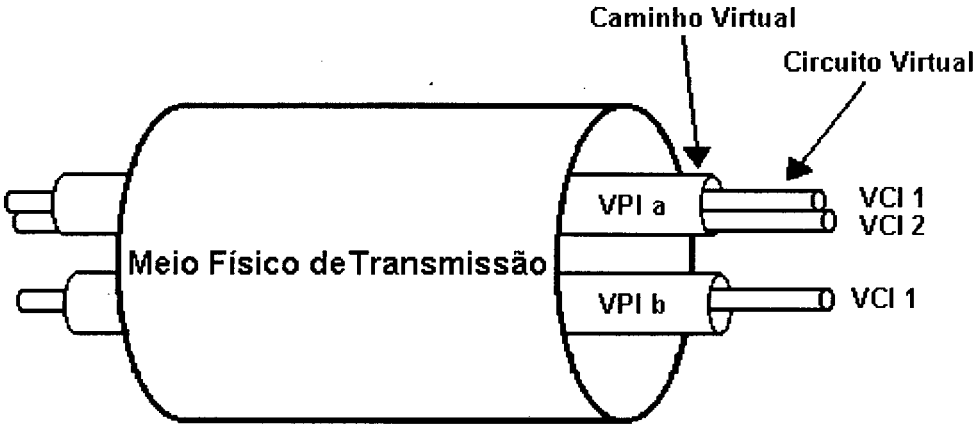

Figura 18: Um enlace, os VPC's (VPI's) e VCC's (VCI's).
Fonte: (SOARES, 1997). Pág. 576.

Tanto um caminho virtual (ou VPC), quanto um canal virtual (ou VCC), através de cada um dos VPL's ou VCL's que respectivamente formam a conexão fim-a-fim, podem ser permanentes ou comutados. As conexões permanentes, também chamadas de estáticas, são estabelecidas com o uso de processos de operação e manutenção de rede, sem a interferência do usuário.

Quando um VCC ou um VPC é comutado, o processo de sinalização faz-se necessário para o estabelecimento e o rompimento do VCC e do VPC. As conexões comutadas, também conhecidas como dinâmicas, são estabelecidas e interrompidas durante o funcionamento da rede.

As estruturas de acesso de RDSI-FE e RDSI-FL possuem canais dedicados para as funções de sinalização, o que constitui uma característica das redes de comutação rápida de pacotes conhecida como *out of band signaling,* que são chamadas **conexões de canais virtuais de sinalização** (ou SVCC, *Signaling Virtual Channel Connection*). Ver Figura 19.

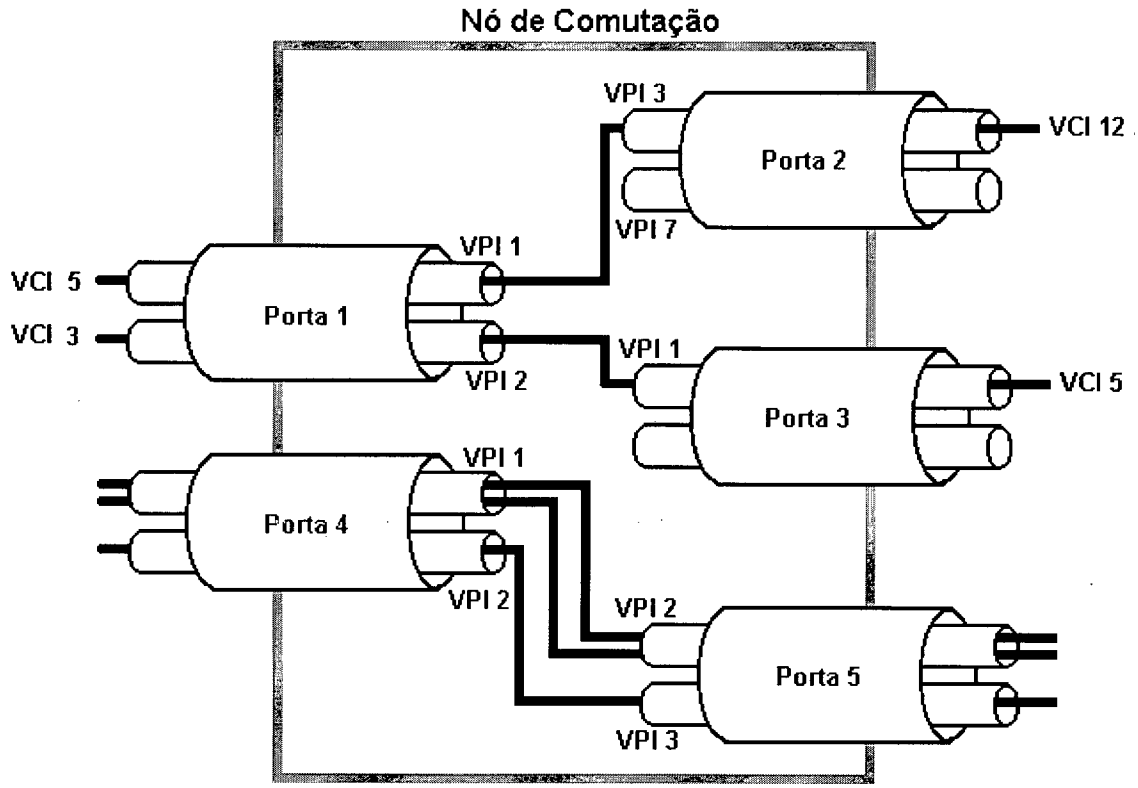

Figura 19: A comutação de portas VCI/VPI.
Fonte: (SOARES, 1997). Pág. 579.

Dentre as funções da camada ATM destacam-se: multiplexação e demultiplexação de células; adição e remoção de cabeçalho de células; interpretação do VCI/VPI; e controle genérico de fluxo – GFC (*Generic Flow Control*).

A "multiplexação e a demultiplexação das células" ocorrem após a chegada e antes da saída de células. Quando as células chegam, elas são de-multiplexadas, do canal para o elemento comutador. Antes de serem retransmitidas são multiplexadas para serem adequadamente transmitidas pelo meio.

A "adição e a remoção do cabeçalho das células" são tarefas imprescindíveis no roteamento da célula. Conforme relatado, a célula traZ em seu cabeçalho a informação de seu VCI e VPI, que é retirada.

A "interpretação de VCI/VPI" é feita de acordo com a tabela de rotas. Um novo cabeçalho é adicionado à célula antes que ela seja encaminhada para uma porta de saída.

O "controle de fluxo genérico (GFC)" trata da preparação das células para transmissão. Com múltiplas células entrando e saindo do elemento comutador concorrentemente, cria-se a possibilidade de conflitos, que devem ser tratados por meio de contenção (bloqueios) ou armazenamento temporário de células.

2.7 A Camada de Adaptação ATM

A camada de adaptação, também conhecida como AAL (*ATM Adaptation Layer*) tem como principal função dar suporte às camadas superiores. Por ser a primeira camada fim-a-fim, a camada de adaptação provê diversos serviços para as aplicações que pretendem trocar informações pela rede, abstraindo detalhes das camadas inferiores. Ver Figura 20.

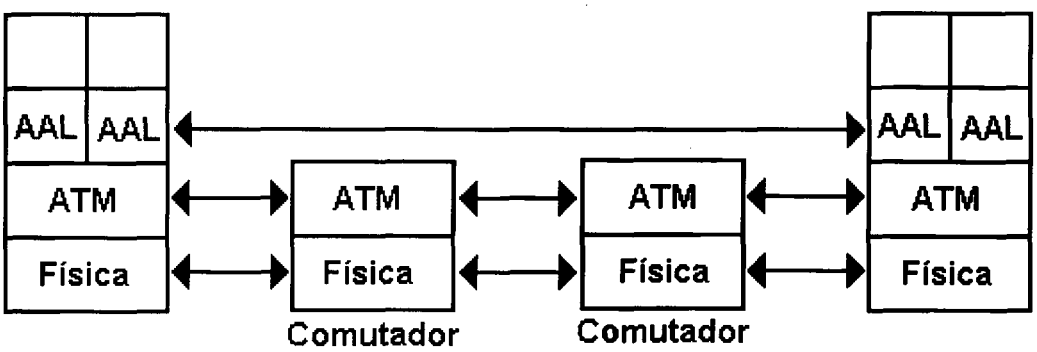

Figura 20: As conexões camada a camada na arquitetura de rede ATM.
Fonte: (SOARES, 1997). Pág. 587.

O ITU definiu quatro classes para diferenciar os serviços providos pela camada AAL, são elas: Classe A, Classe B, Classe C e Classe D. Cada uma das classes possui características distintas, no que diz respeito ao modo de conexão, taxa de geração de bits e na relação temporal entre origem e destino, conforme demonstrado na Tabela 03.

Tabela 03: Classes de serviços definidas pelo ITU na recomendação I.362

Característica	Classe A	Classe B	Classe C	Classe D
Modo de conexão	Orientado a conexão			Sem conexão
Taxa de geração de bits	CBR	VBR		
Relação Tempo	Relacionado		Não relacionado	

Fonte: (SOARES, 1997). Pág. 588.

Dependendo das classes às quais a camada AAL dá suporte, pode-se classificá-la em sete tipos de AAL, a saber: AAL 0, AAL 1, AAL 2, AAL 3, AAL 4, AAL 5 e SAAL. O tipo SAAL (ou *Signaling AAL*) não é utilizado pelo usuário e sim para o transporte da sinalização entre comutadores (NNI) ou entre terminais comutadores e comutadores (UNI).

A camada do tipo AAL 0 simplesmente conecta o usuário aos serviços da camada ATM. A camada do tipo AAL 1 provê basicamente serviços de Classe A, ou seja, orientados a conexão, com taxa de bits constante e com sincronismo reproduzido no destino. A camada do tipo AAL 2 é aquela que provê os serviços de Classe B, orientados a conexão, com taxa de bits variáveis e com sincronismo reproduzido no destino.

As camadas do tipo 3 e do tipo 4 (unidas em um único tipo) fornecem suporte para os serviços de Classe C (tipo 3) com taxas de bits variáveis e sem sincronia e de Classe D (tipo 4) também com taxas de bits variáveis e sem sincronia. As camadas de tipo 3 e 4 diferenciam-se pelo modo de conexão (tipo 3, Classe C sem conexão e tipo 4 , Classe D com conexão).

A camada do tipo 5 visa ser uma simplificação eficiente das camadas do tipo 3 e 4, que apesar de não suportar todas as funções dessas camadas, adicionalmente apresenta a capacidade de multiplexação de conexões.

A camada AAL é dividida em duas subcamadas: a subcamada de Quebra e Remontagem (ou, *Segmentation and Reassembly*); e a subcamada de Convergência (*Convergence Sub-layer*). A **subcamada de Quebra e Remontagem** tem como função

principal a quebra e a remontagem do fluxo de informações em fragmentos a serem transmitidos célula a célula. A **subcamada de Convergência** pode efetuar diversos serviços, a saber: multiplexação de serviços; detecção de perdas de células; e recuperação da ordem das informações.

2.8 O Plano de Usuário

O plano de usuário é utilizado para a transferência de informações dos usuários. Nele estão incluídas: totalmente a Camada Física e a Camada ATM e parcialmente a Camada de Adaptação. Tudo o que acontece na Camada Física diz respeito à transmissão em si, e por isso ela está no Plano do Usuário. A camada ATM, por sua vez, responsável pelo estabelecimento e rompimento das conexões, roteamento das células, controle de erros, etc. também se limita ao Plano do Usuário.

A Camada de Adaptação ATM divide sua função entre o Plano de Usuário e o Plano de Controle. A função de suporte às camadas superiores, com o preparo dos dados para a transmissão, encaixa-se no Plano de Usuário, por caracterizar a transmissão dos dados do usuário.

2.9 O Plano de Controle

O Plano de Controle é aquele no qual os procedimentos de sinalização são necessários para: estabelecer, manter, gerenciar e interromper as conexões, já que estas funções ocorrem no nível da Camada de Adaptação ATM.

Dentre as funções de competência do Plano de Controle destacam-se o controle de admissão de conexões; o policiamento; e o controle de tráfego e de congestionamento. O **Controle de Admissão de Conexões** decide a aceitação ou não do estabelecimento de novas conexões na rede. Tal decisão baseia-se no tráfego atual da rede, nas características e nos recursos solicitados pela nova conexão.

O **Policiamento** é também um controle de tráfego feito sobre as conexões estabelecidas, tendo por finalidade garantir que a taxa de geração de células de cada conexão esteja dentro do limite fixado no momento do estabelecimento da conexão. Quando a geração excessiva de células por parte de uma conexão é detectada, a função do policiamento é decidir (em virtude da ociosidade da rede ou não) pelo descarte das células ou pela alteração da prioridade de descarte das células (CLP).

O **Controle de Tráfego e de Congestionamento** visa controlar ou gerenciar um estado crítico de tráfego, isto é, de congestionamento. O congestionamento é entendido como sendo um estado no qual a rede sofre um tráfego maior do que aquele para o qual ela foi projetada, não conseguindo manter, em virtude disto, a qualidade de serviço assumida no momento em que as conexões foram estabelecidas. O "Controle de Tráfego" é uma abordagem pró-ativa, baseada em uma monitoração que objetiva evitar o congestionamento, e o "Controle de Congestionamento" é uma abordagem reativa a indícios de congestionamento na rede, que tem por objetivo minimizar o impacto do congestionamento sobre o tráfego global da rede.

2.10 O Plano de Gerenciamento

Ao Plano de Gerenciamento são atribuídas diversas funções de gerenciamento, divididas em dois tipos básicos de gerenciamento: o Gerenciamento dos Planos e o Gerenciamento das Camadas. O **Gerenciamento das Camadas** apresenta-se em uma estrutura de camadas, que visa gerenciar os fluxos de informações de operação e de manutenção, assim como os recursos de cada uma das camadas da arquitetura ATM.

O **Gerenciamento dos Planos**, como o próprio nome diz, tem por objetivo gerenciar o plano de usuário, o plano de controle e também o próprio plano de gerenciamento. Esse plano não é estruturado em camadas.

Objetivando o gerenciamento da rede ATM, cinco fases de gerência foram criadas, nas quais, um conjunto mínimo de funções deve ser suportado pela camada física e pela

camada ATM. Com base nestas Funções de Operação e Manutenção (ou seja, *Operation Administration and Maintenance* - OAM) são efetuadas as cinco fases de gerência.

As fases de gerência de rede ATM são: o monitoramento de performance, a detecção de falhas e defeitos, a proteção do sistema, a informação sobre falhas ou desempenho, e a localização de falhas. A fase de **Monitoramento de Performance** é aquela na qual os elementos de rede são monitorados periodicamente, produzindo um conjunto de informações que determinam o "estado" de cada elemento.

A fase de **Detecção de Falhas e Defeitos,** por sua vez, detecta por verificação (contínua ou periodicamente) qualquer mau funcionamento, e como resultado, alarmes são disparados.

Na fase de **Proteção do Sistema,** o elemento gerador de falha é bloqueado com o objetivo de minimizar o impacto do defeito sobre a rede. Quando possível, o elemento gerador da falha é substituído por elementos alternativos.

A fase de **Informação sobre Falhas ou Desempenho** trata os defeitos detectados na rede. Esse tratamento é realizado com a abordagem de propagação das informações sobre os erros para todas as entidades de gerenciamento da rede.

A fase de **Localização de Falhas** tem, como tarefa básica, a determinação exata da localização da falha, quando a informação da falha não seja suficiente para isto.

O sistema de gerenciamento de uma rede ATM deve incluir cinco funções: gerenciamento de configuração, gerenciamento de falhas, gerenciamento de coleta de dados estatísticos, gerenciamento de conexões chaveadas e gerenciamento de conexões permanentes. O **Gerenciamento de Configuração,** diz respeito ao controle operacional dos comutadores, incluindo funções tais como: adição e remoção de módulos de um comutador; habilitação e desabilitação de módulos ou portas do comutador; e detecção de mudanças na configuração.

O **Gerenciamento de Falhas** diz respeito à detecção e ao diagnóstico de falhas ocorridas na rede, que pode ser feita automaticamente ou manualmente, incluindo as seguintes funções: detecção automática de erros (software e hardware); monitoramento de parâmetros críticos; e testes (sobre módulos, portas, enlaces físicos, etc.).

A **Coleta de Dados Estatísticos** diz respeito à detecção e ao diagnóstico de falhas ocorridas na rede, que pode ser feita automaticamente ou manualmente, incluindo dados importantes tais como: taxa de envio por porta de saída; taxa de recepção por porta de entrada; taxa de erro por porta; taxa de descarte por conexão; etc.

O **Gerenciamento de Conexões Chaveadas** tem por objetivo permitir uma adequada tarifação das conexões, gerando para cada conexão rompida, dados relativos à conexão em si, e detalhando informações durante o tempo no qual a conexão foi mantida.

O **Gerenciamento de Conexões Permanentes,** monitora a operação e o tráfego destas conexões, verificando o adequado funcionamento e efetuando o restabelecimento da conexão, caso esta tenha sido rompida.

2.11 A operação de uma rede ATM

A operação ou o funcionamento de um comutador ATM é relativamente simples, cada comutador possui um determinado número portas (de entrada e de saída) e uma tabela (*Local Translation Table* - LTT) que determina os VCI ou VPI conectados a cada porta.

As redes ATM suportam dois tipos de conexões distintas: as **conexões virtuais permanentes** PVC (*Permanent Virtual Connections*) e as **conexões virtuais comutadas** SVC (*Switched Virtual Connections*).

Uma conexão virtual permanente (PVC) é uma conexão entre dois ou mais comutadores ATM (da origem ao destino) que define um enlace lógico. O estabelecimento e a configuração são efetuados por algum dispositivo externo, normalmente de uma ferramenta de gerência de rede, com auxílio manual.

As conexões virtuais comutadas (SVC) são automaticamente configuradas através do uso de um protocolo de sinalização, dispensando portanto intervenção manual. Por serem mais fáceis de usar, são mais utilizadas.

O processo de sinalização ATM, para a configuração de uma SVC, é iniciado por um sistema de borda, no qual um usuário solicita uma conexão ou um serviço que demande uma conexão que é estabelecida até o destino. Os pacotes de comunicação são transmitidos através de um enlace, ou canal virtual, dedicado a esta tarefa.

Capítulo 03: Gerência de Redes

Capítulo 03: Gerência de Redes

Toda e qualquer rede de computadores como estrutura computacional completa, precisa ser gerenciada. Tal gerenciamento torna possível garantir a sua disponibilidade, a qualidade de seus serviços, e o funcionamento da rede num nível de desempenho aceitável.

As aplicações de gerência de rede têm vários objetivos, destacando-se o suporte para que a rede possa estar disponível o maior tempo possível, as ferramentas para monitorar tráfego, desempenho e qualidade de serviço, permitindo assim que a tomada de decisão possa ser feita, antes que problemas aconteçam.

Já, o gerenciamento numa rede ATM, é estruturado da mesma maneira como é estruturada a maioria dos sistemas de gerência de comunicação. Ele está baseado em três elementos: os agentes; o gerenciador de caixa e o sistema de gerenciamento da rede. Ver Figura 21.

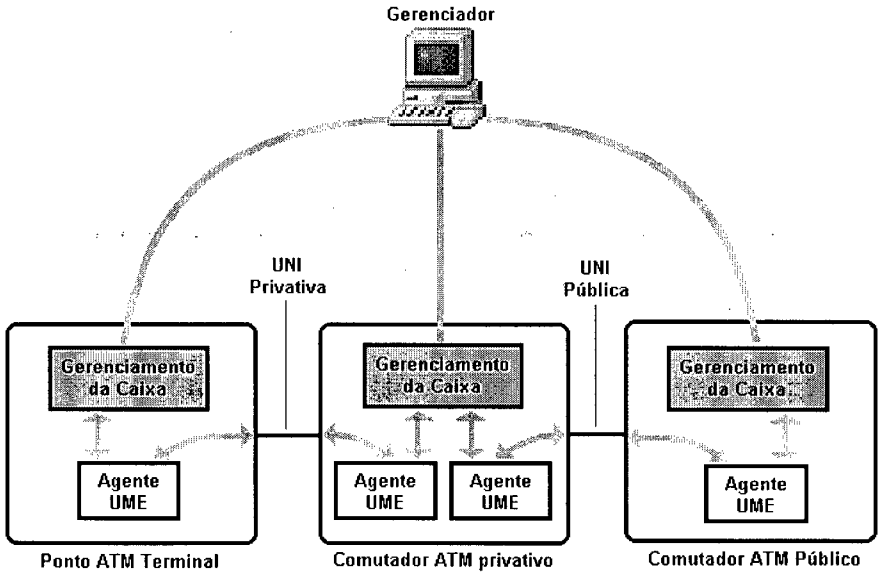

Figura 21: A estrutura de gerenciamento ATM.
Fonte: (SOARES, 1997). Pág. 629.

Conforme Soares (SOARES, 1997), os **Agentes** são entidades espalhadas pelo sistema de comunicação da rede, e localizados junto aos elementos ativos. Esses agentes

são chamados de ***UME*** - Entidades de Gerenciamento UNI (ou seja, *UNI Management Entities*).

Gerenciadores de Caixa, que monitoram o interior dos comutadores, a maioria dos comutadores suporta interfaces nas quais podem ser conectados os gerenciadores de caixa, para a monitoração e controle.

O **Gerenciador de Sistema** se comunica com os comutadores espalhados pelo sistema de comunicação, oferecendo uma forma de monitoração e operação centralizada para a gerência da rede.

Segundo Soares (Soares, 1997), os detalhes relacionados ao funcionamento das entidades no **Sistema de Gerenciamento de Rede** não foram ainda especificados pelo ITU. Assim sendo, foi desenvolvida uma interface prévia para uso temporário chamada ILMI.

A **Interface Interina de Gerenciamento Local** (*Interim Local Management Interface* – ILMI), foi concebida para ser uma base padronizada de informações de gerenciamento, definindo as informações que devem ser monitoradas e/ou controladas pelos agentes.

A ILMI utiliza o **Protocolo de Gerenciamento de Rede Simples** – SNMP (ou *Simple Network Management Protocol*) como meio de regular a troca de informações entre os agentes, ou entidades UME até o Gerenciador do Sistema. As informações estão descritas em uma **Base de Informações de Gerenciamento** – MIB (Management Information Base).

As informações de gerenciamento da Interface Rede-Usuário (UNI) descritas na MIB, podem ser categorizadas em seis classes distintas: informações da camada física; informações da camada ATM; informações estatísticas da camada ATM; informações

sobre as conexões VPC; informações sobre as conexões VCC e informações de registro de endereços.

As **Informações da Camada Física** definem cada interface do comutador (Controladores de Entrada - CE's e Controladores de Saída - CS's) e a estrutura de transmissão (e.g. tabelas de roteamento, status das portas, etc.) que refletirão as conexões físicas da rede.

As **Informações da Camada ATM** trazem as informações que dizem respeito a todas as Conexões Virtuais (VC) e a todos os Caminhos Virtuais (VP) configurados em cada uma das interfaces do comutador.

As **Informações Estatísticas da Camada ATM** acumulam dados estatísticos relativos a todas as conexões (VCC e VPC) de uma interface (e.g. total de células transmitidas ou recebidas, total de células com erros, total de células descartadas, etc.).

As **Informações sobre Conexões VPC** e as **Informações sobre Conexões VCC** aglutinam o mesmo tipo de informação, relacionado a Caminhos Virtuais (VPC) e a Conexões Virtuais (VCC), respectivamente. Estas informações podem descrever como é o tráfego no respectivo VPC ou VCC, com base em informações do tipo: taxa máxima de geração de células; taxa média de geração de células; o estado operacional da conexão; etc.

As **Informações sobre Registros de Endereços** permitem o registro de pontos terminais de rede (ou NT, *Network Termination)* em um dado comutador.

A alta velocidade e a capacidade que as redes ATM possuem de manipular diferentes tipos de dados, determinam uma necessidade maior de controle e monitoração na rede (ALEXANDER, 1995).

3.1 Gerência de Redes com SNMP

Segundo Krishnan e Fuller (KRISHNAN, 1997), a chave para uma plataforma aberta de gerenciamento de rede está baseada em dois aspectos: no uso de um **Protocolo de Comunicação Comum** e num conjunto padrão de **Objetos Gerenciáveis** ou de **Base de Informações de Gerenciamento**.

Cada elemento gerenciável mantém uma ou mais informações, armazenadas individualmente em variáveis, que permitem descrever o seu estado. Estas variáveis são conhecidas como objetos gerenciáveis. E o conjunto de todos os possíveis **objetos gerenciáveis**, ou seja, variáveis com informação (atributos), apresenta-se como uma estrutura de dados numa hierarquia em árvore conhecida como Base de Informações para Gerenciamento (ou MIB, *Management Information Base*).

A MIB é um arquivo que contém uma hierarquia de objetos do equipamento gerenciável, bem como o nome (ou, *Object ID*), a sintaxe e os privilégios de acesso para cada variável. É importante perceber que somente os tipos de objetos de um determinado equipamento gerenciável estão especificados na MIB. Não sendo portanto, especificados os objetos ou suas instâncias, conforme destaca a Adventnet (ADVENTNET, 1998).

Na MIB II (RFC-1213), por exemplo, ***ifinOctets*** especifica o tipo do objeto que armazena o número de octetos que entraram em uma interface. Uma interface particular pode ser especificada como ifInOctets 1, ifInOctets 2, ifInOctets 3,... Dependendo do número dessa interface.

Considerando que os ambientes de rede são predominantemente heterogêneos, surge como necessidade, portanto, a especificação rígida de seus elementos. Essa especificação deve incluir, tanto a MIB, que é mantida pelos diversos elementos de uma rede, quanto o protocolo de comunicação, utilizado para a troca de pacotes de rede com informações de gerência entre os elementos da mesma.

Tanto para o protocolo SNMP quanto para o CMIP, encontram-se várias MIB's definidas, por diversas organizações de padronização, a saber: ATM Fórum (SNMP e CMIP); IETF ou *Internet Engineering Task Force* (SNMP); NM Forum (CMIP); e ITU (CMIP); além das MIB's proprietárias desenvolvidas pelos fabricantes para gerenciar seus equipamentos de rede (baseadas principalmente em SNMP).

Diferentes tipos de equipamentos estão associados a objetos específicos. Assim, para que o gerente ou a aplicação de gerência possa operar inteligentemente com os dados disponíveis no equipamento, o gerente precisa saber os nomes e os tipos dos objetos associados a um equipamento gerenciável.

Apesar de terem objetivos específicos, o conteúdo das MIB's das diferentes organizações sobrepõem-se algumas vezes. Algumas vezes pode ser necessário utilizar várias MIB's para que um objetivo da gerência seja alcançado. Por isto, é importante compreender as MIB's, sabendo os seus propósitos, seus escopos e seus inter-relacionamentos para gerenciar as redes ATM (KRISHNAN, 1997).

O protocolo escolhido para implementação e estudo foi o protocolo SNMP, porque a grande maioria dos elementos de rede gerenciáveis o suportam, podendo ser considerado atualmente um padrão de mercado.

O modelo de gerência de rede SNMP é baseado em quatro componentes: nodos gerenciados; estação de gerenciamento; informações de gerenciamento e um protocolo de gerenciamento. Ver Figura 22.

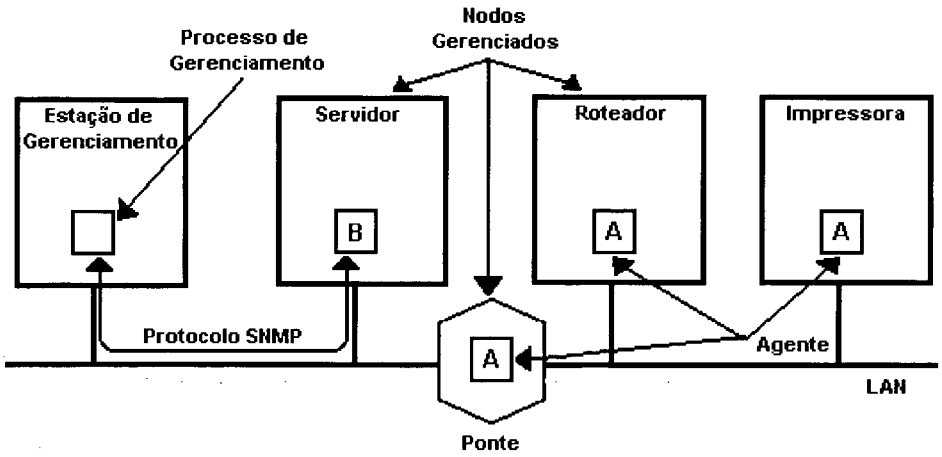

Figura 22: Componentes do modelo de gerenciamento SNMP.
Fonte: (TANENBAUM, 1997). Pág. 631.

Os **Nodos Gerenciados** são os elementos ativos da rede (e.g. roteadores, servidores, comutadores, etc.) ou são segundo Tanenbaum (TANENBAUM, 1997), qualquer dispositivo capaz de prover informações de seu *status* para o mundo externo. De acordo com o modelo de gerenciamento SNMP, para que um nodo de rede possa ser gerenciado faz-se necessário que o mesmo execute um processo de gerenciamento SNMP chamado **Agente SNMP**. Cada agente mantém uma base local de dados que pode descrever o seu estado, o seu histórico ou até mesmo afetar a sua operação, etc.

A ***Estação de Gerenciamento*** é o componente central do modelo de gerenciamento de rede, na qual computadores, através de um software específico, executaram a gerência da rede, num programa conhecido como **Processo de Gerência**. A estação de gerenciamento, ou simplesmente o **Gerente** executa um ou mais Processos de Gerenciamento, os quais se comunicam com os agentes para obter as informações que possam permitir uma tomada de decisão. Assim sendo, a gerência do processo centraliza-se no Gerente.

Para tanto o protocolo SNMP permite a interação do gerente com o agente. Através de requisições, o gerente pode verificar ou alterar o estado dos objetos gerenciáveis nos agentes, pois o SNMP baseia-se fortemente na comunicação do tipo pergunta e resposta (ou seja, *query-response*).

A exceção a esta característica interativa de pergunta e resposta se dá quando eventos não planejados acontecem. Desde um defeito num elemento de rede até o congestionamento na rede em si, cada evento considerado significativo deve estar definido na MIB.

A comunicação SNMP permite essencialmente quatro tipos de operações entre gerentes e agentes (equipamentos gerenciáveis):

- o gerente pode executar um *get* (ou seja, fazer uma leitura) para obter informações do agente sobre um atributo de um objeto gerenciado;
- o gerente pode executar um *get-next* para fazer um *get* em um próximo objeto na árvore de objetos do equipamento gerenciado;
- o gerente pode executar um *set* (ou seja, fazer uma gravação) para alterar o valor de um atributo de um objeto gerenciado; e
- o agente pode enviar um *trap*, ou seja, notificação assíncrona, para o gerente fazendo indicação sobre algum evento no equipamento gerenciado.

Quando um agente tem a informação de que um evento significativo ocorreu, esta informação é repassada para todas as estações de gerenciamento incluídas em sua lista de configuração. Este procedimento é conhecido como *Trap*.

Nem sempre o *trap* especifica qual o evento ocorrido, portanto, nesses casos, torna-se interessante que o gerente execute o *Polling*. O *polling* realiza uma apuração, através de pergunta-resposta, junto a todos os nodos gerenciados para verificar se eventos inesperados ocorreram, somente como precaução. O modelo que utiliza o *polling* a intervalos longos de tempo, com uma aceleração quando são reportados eventos inesperados através de *trap's* e é conhecido como *Trap Directed Polling*.

Portanto, no modelo de gerenciamento por *polling,* o Gerente busca as informações gerenciais junto aos Agentes SNMP's, através de uma comunicação

repetida a intervalos de tempo configuráveis, usando o protocolo de gerenciamento SNMP.

Desta forma, o Gerente centraliza as funções de armazenamento de informação, de tomada de decisão e o controle do processo de gerência. Os agentes ficam distribuídos pela rede. Eles são responsáveis pela coleta das informações, monitoração e geração de alertas. A Tabela 06, mostra os diferentes tipos de mensagens utilizadas no contexto do protocolo SNMP.

Tabela 04: Tipos de Mensagens SNMP.

Mensagem	Descrição
get-request	Requisição do valor de uma ou mais variáveis
get-next-request	Requisição do valor da variável seguinte
get-bulk-request	Requisição dos valores de uma tabela de variáveis
set-request	Altera o valor de uma ou mais variáveis
inform-request	Mensagem Gerente-para-Gerente descrevendo a MIB local
SnmpV2-trap	Mensagem de aviso do Agente-para-Gerente

Fonte: (TANENBAUM, 1997). Pág. 643.

Os objetos gerenciados são especificados, com **identificadores de objetos** (object ID's, ou simplesmente ID's) que são uma série de números que univocamente identificam um objeto para um agente SNMP.

Para enviar mensagens SNMP do tipo *get* e *set* para um agente, são necessárias duas informações: o identificador do objeto (*Object ID*), e a instância (objeto de um tipo de dado específico).

3.2 Gerência de Redes via WWW

Não é novidade que o uso da Internet para todas as finalidades possíveis, tem sido cada vez maior. Porém, no ambiente corporativo, esta nova plataforma de trabalho passou a ser difundida a ponto de terem sido criados os termos: Intranet e Extranet.

A Intranet e a Extranet definem ambientes computacionais privados, baseados na arquitetura de rede Internet (TCP/IP), com um servidor ***World Wide Web*** separado da Internet pela segurança de um *firewall* ou por técnicas de tunelamento. Estes ambientes podem âmbito restrito para uso interno da empresa (no caso da Intranet) ou para uso na interconexão de unidades remotas da empresa (no caso da Extranet).

Por suportar SNMP (ou mesmo CMIP) este ambiente permite aos administradores de rede usar as Intranet's (ou Extranet's) para monitorar e manter a rede, a fim de gerenciar a rede da empresa, com todas as funcionalidades oferecidas pelo ambiente *Web*.

Estas funcionalidades permitem aos administradores da rede rapidamente e com facilidade configurar, controlar, e acessar individualmente os elementos de rede, a partir de qualquer computador (conectado à rede) independentemente da plataforma e através de qualquer navegador *Web*.

Conforme Hyde (HYDE, 1998) o Gerenciamento Baseado na *Web* (ou WBM, *Web-Based Management*) é uma solução revolucionária para o gerenciamento de rede, que permitirá transformar a maneira pela qual os usuários gerenciam suas redes. Essa solução que pode parcialmente ser atribuída à popularidade das Intranet's (CERUTTI, 1998).

Além de oferecer aos administradores de rede um ambiente multi-plataforma e distribuído para o gerenciamento, o gerenciamento baseado na *Web* permite que os usuários utilizem-se das informações e alertas de gerenciamento, já que a facilidade de uso da interface baseada em páginas *Web* permite que qualquer usuário, sabendo navegar na Internet, possa buscar informações de gerência sobre o estado da rede e de seus elementos, além de diminuir o custo de treinamento de usuários.

Esta proposta torna-se mais facilmente aceita, quando se verifica que os programas tradicionais de gerência de rede possuem mecanismos relativamente

complexos de instalação, configuração, e utilização. Apresentando um custo elevado de aquisição (BAROTTO, 1998), sem alcançar a flexibilidade do Gerenciamento Baseado na *Web*.

Esta mudança no ambiente computacional corporativo traz reflexos nas mais diversas áreas. O modelo tradicional Cliente/Servidor pode ser repensado para otimizar o uso da rede e reduzir o custo de desenvolvimento e de suporte.

Com base na arquitetura de rede Internet, o gerenciamento SNMP pode vir a ser implementado com muitas tecnologias. A tecnologia mais comum, segundo Hyde (HYDE, 1998) é a linguagem HTML (ou seja, *Hypertext Markup Language)*, criada para desenvolver páginas de hiper-texto e que é responsável, até hoje, por muito do que se vê na *Web*.

Ao longo dos anos, a linguagem HTML evoluiuao longo dos anos evoluiu para suportar novas tecnologias, sempre com o objetivo de estender a capacidade do ambiente *Web*. Como exemplo, tem-se: o CGI (ou seja, *Common Gateway Interface*) desenvolvido para suportar acesso a bancos de dados via *Web*; o *Applet Java* (programas Java executáveis por um navegador *Web*) concebido para permitir o desenvolvido e a execução de programas via *Web*; etc.

3.3 Gerência de Redes com Java

Um código em linguagem de programação Java deve ser "interpretado" no momento da execução por um programa específico (conhecido como interpretador). E em função disto seu código pode ter execução mais lento do que um código desenvolvido em uma linguagem compilada (que gera um código binário executável). Em contrapartida, enquanto o código binário compilado pode rodar somente numa plataforma, o código interpretável pode vir a ser executado em qualquer plataforma que tenha o respectivo interpretador instalado.

Considerando que a proposta da linguagem Java é executar os seus programas dentro do ambiente *Web*, foi feito um esforço junto às empresas que desenvolvem navegadores (que são clientes no ambiente *Web*) para que estes mesmos trouxessem embutido um interpretador Java, habilitando-os desta maneira a executar *Applet's Java* nas mais diversas plataformas.

O código gerado pelo interpretador Java é chamado de *Java Code* e para que ele possa ser executado num outro computador, deve haver neste um interpretador chamado **JVM** (ou seja, *Java Virtual Machine*). Existem diferentes versões de JVM's para as mais diferentes plataformas, além de estarem embutidos nos navegadores *Microsoft Internet Explorer* e *Netscape Navigator*.

Sendo assim, um programa em Java pode ser desenvolvido para ser executado em computadores isolados, como qualquer aplicação, ou para ser executado a partir de uma página HTML, como um *applet*. A diferença entre um *Applet* e uma Aplicação, tecnicamente falando, é que os applet's possuem restrições de segurança impostas pelos navegadores. Estas restrições impedem o applet de acessar recursos locais (tais como: memória, unidades de disco, etc.) e limitam o seu acesso aos recursos de rede. Desta maneira, os *applet's* tornam-se mais seguros, minimizando riscos e permitindo executá-los sem quebrar a segurança da rede.

A necessidade de utilizar-se de *applet's* está no fato de que o HTML puro não permite o desenvolvimento de programas avançados, com recursos de gráficos dinâmicos, etc. Além do que, *applet's* Java podem apresentar informações em tempo real atualizadas através de *polling* e *traps*.

3.4 Gerência de Redes com AdventNet Management Builder

AdventNet Management Builder é uma ferramenta, desenvolvida em Java, para a implementação de aplicações de gerenciamento de equipamentos de rede. Essa ferramenta utiliza-se de um gerador de *Java Code* composto por um ambiente de

desenvolvimento gráfico e uma biblioteca de componentes SNMP, gráficos e de gerenciamento. Ver Figura 23.

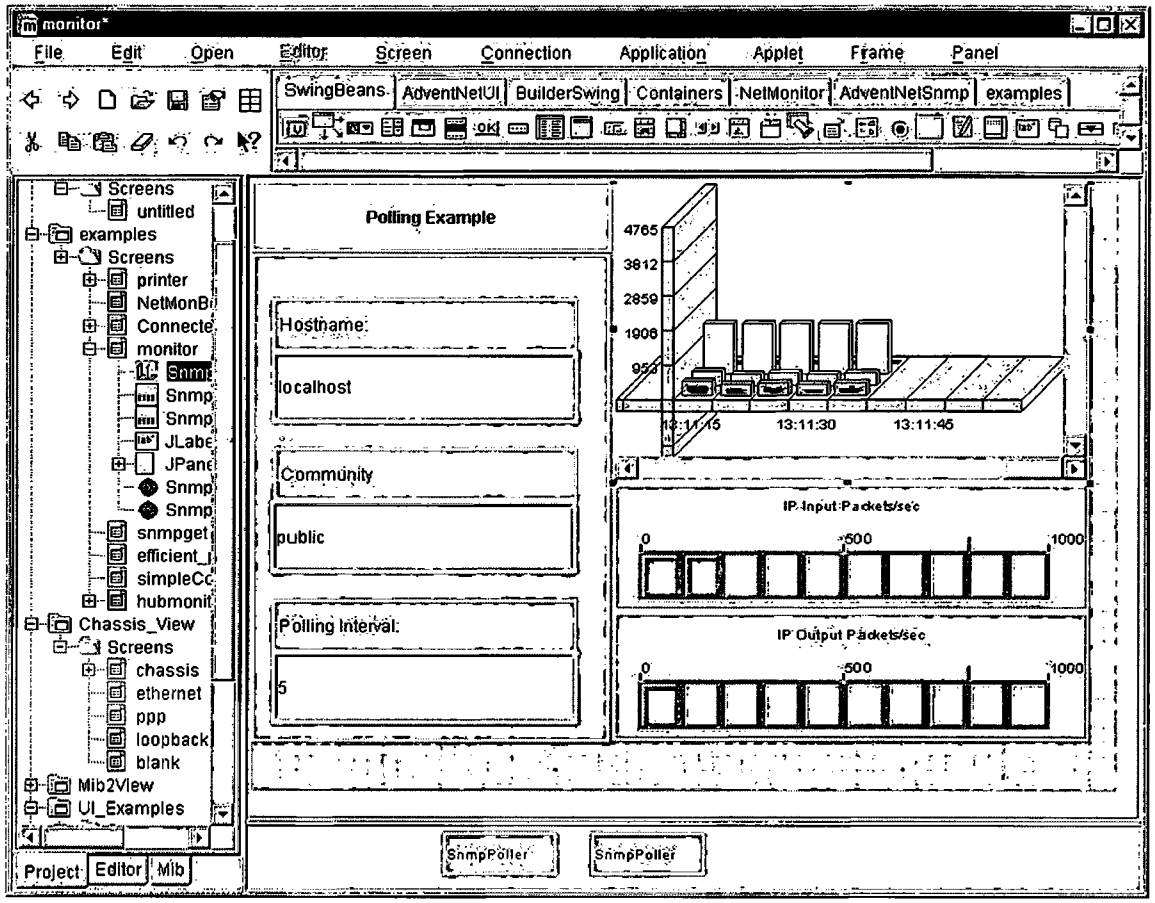

Figura 23: A interface do Adventnet Management Builder.

A exemplo das ferramentas de desenvolvimento visual, o *Management Builder* apresenta uma interface na qual é possível "montar a tela" do programa que se deseja implementar. Utilizando-se para isto da sua biblioteca de componentes, que vão desde botões, rótulos, painéis, etc. até componentes SNMP.

Além de criar a interface gráfica, o *Management Builder* permite que o usuário implemente código livremente, através dos eventos existentes em cada componente da biblioteca, tal qual em outras ferramentas de desenvolvimento. Também é possível criar uma ligação de componentes que permite a troca de dados entre os mesmos.

O *Management Builder* permite que o usuário faça opção entre implementar um applet e um aplicativo Java. Ressalta-se que, ao desenvolver um applet, o teste de execução deve ser efetuado dentro de um navegador *Web,* e para tanto, o Management Builder possui um Servidor *Web* (*Web Server*) e um Servidor Applet SNMP (*SAS*) embutidos.

Capítulo 04: O Ambiente de Desenvolvimento

Capítulo 04: O Ambiente de Desenvolvimento

O ambiente no qual desenvolveu-se este trabalho foi a rede, da Universidade Federal de Santa Catarina – UFSC, principalmente, por que esta constitui-se num ambiente tecnologicamente adequado para pesquisa na área de redes ATM.

A Universidade Federal de Santa Catarina encontra-se inserida na sociedade catarinense por uma integração que começou em 1932. No contexto da área tecnológica, e mais especificamente da computação, esta realidade também se apresenta por várias razões.

A primeira destas razões e talvez a mais visível delas, é a formação de profissionais, ou seja, o ensino, pois na UFSC são formados profissionais da área da computação desde 1980. A formação de profissionais – de todas as partes do estado, já seria suficiente para tornar importante a participação da UFSC na sociedade catarinense. Uma universidade porém, deve integrar-se a sua sociedade também através da pesquisa e a UFSC não se furtou desta incumbência.

A partir da criação do Curso de Pós-graduação em Ciência da Computação, em 1987, passaram a ser desenvolvidas mais fortemente as pesquisas. No início, com um o Programa de Especialização, posteriormente substituído por um Programa de Mestrado que é atualmente oferecido juntamente com o Programa de Doutorado.

4.1 A Universidade Federal de Santa Catarina - UFSC

A primeira Instituição de Ensino Superior no Estado de Santa Catarina, iniciou sua história em 1932, com a criação de diversas Faculdades. No ano de 1960, uma reforma universitária extinguiu as Faculdades e criou a **Universidade Federal de Santa Catarina - UFSC**. A Ciência da Computação foi introduzida na UFSC pelos projetos de pós-graduação da área tecnológica, no ano de 1970.

O primeiro computador da UFSC foi instalado em 1970, era um equipamento IBM 1130 e foi o segundo computador instalado em Santa Catarina (NPD, 1999) e (SANTOS, 1996). Este computador ficou vinculado ao então recém-criado **Departamento de Ciências Estatísticas e da Computação**.

Em 1976, houve uma separação nas atividades de computação na Universidade. Ao **Departamento de Ciências Estatísticas e da Computação** foram incumbidas as funções de ensino da computação. E foi criado um novo departamento chamado de **Departamento de Processamento de Dados**, para tratar prioritariamente com as atividades de administração universitária.

No primeiro semestre de 1977, iniciaram-se as atividades do Curso de Bacharelado em **Ciência da Computação**. Em 1987, foi criado o **Programa de Pós-Graduação** em Ciência da Computação da UFSC. Esse programa iniciou suas atividades com um curso em nível de especialização. Em 1992, foi instalado o **Programa de Mestrado** e em 1999 o **Programa de Doutorado** do Curso de Pós-graduação em Ciência da Computação.

4.2 O Núcleo de Processamento de Dados - NPD

O Departamento de Processamento de Dados, ao ser criado em 1976, fica vinculado diretamente à Reitoria e com atividades voltadas para o ensino, pesquisa e extensão, mas com prioridade para aplicações na própria administração universitária. E tinha as seguintes atribuições:

- análise, programação e operação de sistemas para a universidade;
- atendimento aos usuários do computador;
- processamento de programas de alunos, professores e pesquisadores da UFSC;
- manutenção e desenvolvimento de sistemas operacionais.

Em 1977, a UFSC adquire um computador IBM 360-40, com o intuito de ampliar não só o atendimento à pós-graduação e à pesquisa, como também às aplicações administrativas da universidade. Em 1980 foi comprado um IBM 4341. Após este período de franca expansão (1976-1980), o setor de informática passa por uma fase de "crise" (NPD, 1999).

Somente em 1985, quando o então Departamento de Processamento de Dados passa a ser o **Núcleo de Processamento de Dados** – NPD, novos investimentos são feitos (NPD, 1999).

Em 1988, cria-se uma comissão técnica de trabalho para analisar e direcionar os investimentos de recursos obtidos junto ao FINEP, que resulta na abertura de uma licitação internacional com o objetivo de iniciar a implantação do modelo de processamento distribuído na UFSC, o qual deu início a **redeUFSC**.

4.3 A Rede Nacional de Pesquisas - RNP

No momento de implantação da redeUFSC são formados autonomamente, no Brasil, alguns embriões de redes, ligando grandes universidades e centros de pesquisa do Rio de Janeiro, de São Paulo e de Porto Alegre aos Estados Unidos.

Objetivando integrar esses esforços, e coordenando uma iniciativa nacional em redes no âmbito acadêmico, o Ministério da Ciência e Tecnologia formou um grupo para discutir o tema. O grupo era composto por representantes do:

- CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico
- FINEP - Financiadora de Estudos e Projetos
- FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo
- FAPERJ - Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro e
- FAPERGS - Fundação de Amparo à Pesquisa do Estado do Rio Grande

do Sul.

O resultado é o projeto da RNP, formalmente lançado em setembro de 1989, que tem sua atuação limitada ao âmbito federal (interestadual) e internacional. Fica definido que iniciativas de redes estaduais seriam estimuladas para a ampliação da capilaridade da rede (RNP, 1999).

Nesse ano, a UFSC ingressou nas redes internacionais, mais especificamente a BitNet, quando o NPD abriu uma conta de acesso discado à FAPESP. Este sistema funcionou por aproximadamente um ano, migrando posteriormente para a RENPAC. Inicialmente de maneira discada e só então de maneira dedicada (SANTOS, 1996).

No ano de 1991 foi criado o Programa Prioritário do MCT - Ministério da Ciência e Tecnologia, apoiado e executado pelo CNPq -, cuja missão principal é operar um serviço de *backbone* Internet voltado à comunidade de ensino e de pesquisa do Brasil chamado Rede Nacional de Pesquisa – RNP.

A partir de então, a RNP introduziu a tecnologia Internet no país, desempenhando um papel de destaque na consolidação do *backbone* nacional para a comunidade acadêmica, tanto na disseminação de serviços e aplicações de rede, quanto na capacitação de recursos humanos.

A RNP opera um serviço de *backbone* para atender à comunidade acadêmica e de pesquisa, oferecendo acesso à Internet através dos seus Pontos de Presença regionais. Os PoP's da RNP, que compõem o seu *backbone* nacional, estão presentes em várias cidades do país, principalmente nas capitais. Ver o *backbone* atual da RNP na Figura 24.

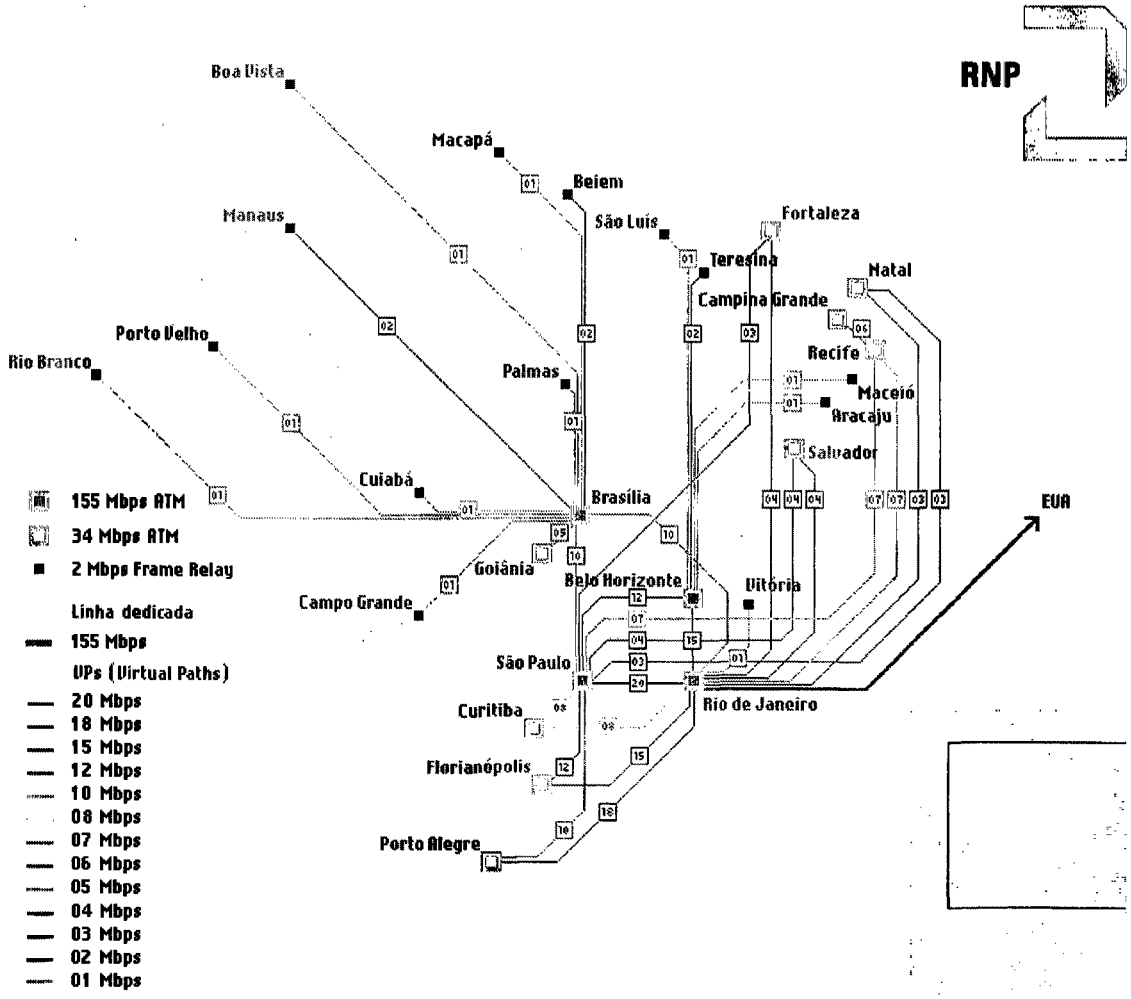

Figura 24: O *backbone* da RNP.
Fonte: (RNP, 1999).

Os Pontos de Presença atendem às necessidades operacionais da rede, à demanda de conectividade, à demanda de informações aos usuários de todo o Estado. Além disso, tem a função de coordenar e operar serviços Internet (POP-SC, 1999). O PoP-SC da RNP, encontra-se em Florianópolis, fisicamente localizado no NPD da UFSC.

4.4 A Rede de Ciência e Tecnologia - RCT

O surgimento da Rede de Ciência e Tecnologia de Santa Catarina RCT-SC, é resultado de um processo que teve início em 1991, com a apresentação de uma proposta à Funcitec, visando à implantação da RCT. Esta proposta foi feita pela UFSC, e coordenada pelo NPD – através de seu Diretor Técnico o Sr. Edison T. L. Melo, em

conjunto com o Departamento de Informática e Estatística – através da Profa. Elizabeth Specialski (SANTOS, 1996).

Mais tarde, em 1992 esta proposta foi revisada. Em 1993, foram reunidas diversas entidades, para discutir a montagem de um "sistema de informação em ciência e tecnologia" para o estado (SANTOS, 1996).

Em 1994, com base nessas discussões, a UFSC, a UDESC (Universidade do Estado de Santa Catarina), a ACAFE (Associação Catarinense das Fundações Educacionais) e a EPAGRI (Empresa de Pesquisa e Agropecuária e Difusão de Tecnologia de Santa Catarina) apresentaram uma nova proposta para a rede estadual acadêmica, que foi então adotada pelo Governo (SANTOS, 1996).

Os esforços são recompensados, pois o projeto da RCT sai do papel em 1995, e então é firmado – conforme (FUNCITEC, 1999), um convênio entre o governo do Estado de Santa Catarina, através do FUNCITEC, com:

- Universidade Federal de Santa Catarina –UFSC;
- Universidade do Estado de Santa Catarina – UDESC;
- Associação Catarinense das Fundações Educacionais – ACAFE;
- Serviço de Apoio à Pequena e Média Empresa – SEBRAE;
- Empresa de Pesquisa e Agropecuária e Difusão de Tecnologia de Santa Catarina –EPAGRI e;
- Federação das Indústrias do Estado de Santa Catarina –FIESC.

A RCT-SC é uma extensão estadual da Rede Nacional de Pesquisa – RNP e da INTERNET, e constitui a infra-estrutura básica do Sistema Estadual de Informação em Ciência e Tecnologia. O atual backbone da RCT é mostrado na Figura 25.

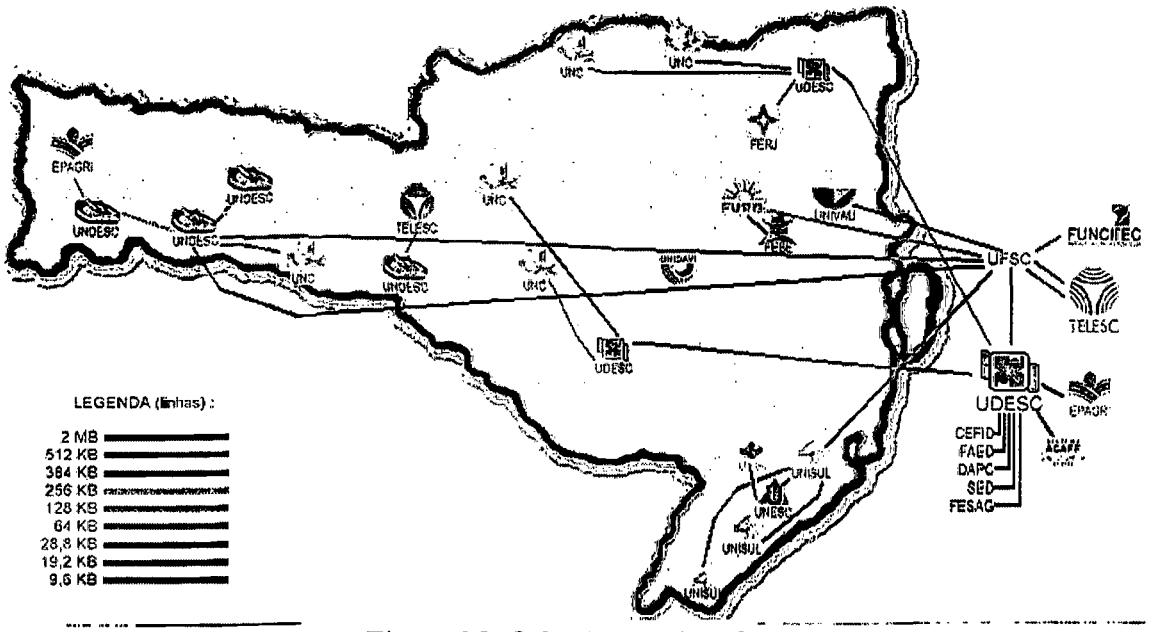

Figura 25: O *backbone* da RCT.
Fonte: (FUNCITEC, 1999).

Além de ser um PoP-RCT, coube também ao NPD da UFSC à administração do *backbone* da RCT (POP-RCT, 1999), dada a experiência e o conhecimento de sua equipe. Por ser o PoP-SC da RNP e por administrar a RCT-SC, o NPD da UFSC passou a ser reconhecido em sua elevada importância no ácesso à Internet – dentro do contexto educacional no estado de Santa Catarina. Fortalecendo-se, também, como um ambiente heterogêneo e propício para pesquisas na área de redes de computadores no âmbito nacional.

Tal percepção pode ser medida pela escolha da UFSC como um dos centros de experimentação e pesquisa (ao lado da PUC-Rio e COPPE/UFRJ), no convite para "Propostas de Utilização de Plataformas de Alta Velocidade", que foi organizado pelo Comitê Internet/BR e pelo Grupo de Trabalho de Pesquisa e Desenvolvimento em Redes (GT P&D, 1997).

Ou ainda, pela aprovação do projeto da Rede Metropolitana de Alta Velocidade de Florianópolis – REMAV-FLN, em resposta ao edital da RNP/ProTeM-CC (Programa Temático Multi-Institucional em Ciência da Computação) (PROTEM, 1998) e (PROTEM, 1999).

4.5 A Rede Metropolitana de Alta Velocidade de Florianópolis - REMAV-FLN

O projeto da Rede Metropolitana de Alta Velocidade de Florianópolis – REMAV-FLN, foi submetido à chamada de projetos ProTeM/RNP e aprovado. O objetivo principal do projeto é colocar em operação e gerenciar a Rede Metropolitana de Alta Velocidade de Florianópolis.

Criando um ambiente heterogêneo – composto pela UFSC e por outras instituições e empresas, que possibilita testar e disponibilizar experimentos de aplicações interativas na rede metropolitana ATM, o projeto REMAV-FLN, também permite capacitar as pessoas e as instituições envolvidas a operar e gerenciar redes ATM de alta velocidade. A topologia da REMAV-FLN é mostrada na Figura 26 abaixo.

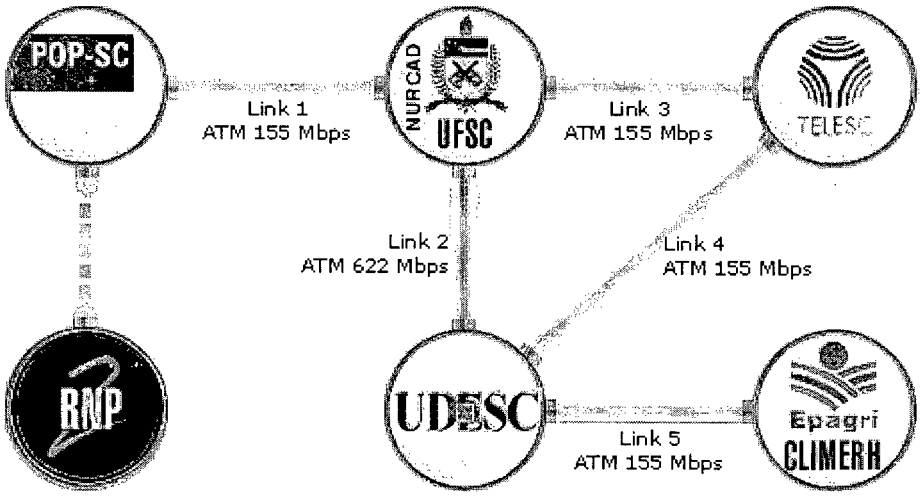

Figura 26: O *backbone* da REMAV-FLN.
Fonte: (REMAV-FLN, 1999).

As instituições que participam do projeto são:

- Universidade Federal de Santa Catarina - UFSC
- Universidade do Estado de Santa Catarina – UDESC
- Empresa de Telecomunicações de Santa Catarina – TELESC
- Centro Integrado de Meteorologia e Recursos Hídricos de Santa

Catarina/Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina (CLIMERH/EPAGRI)

O Núcleo de Processamento de Dados – NPD da UFSC, que foi criado em 1977 com o objetivo de ser um *bureau* de processamento, foi ao longo dos anos adaptando-se às novas realidades, agregando muitos egressos da própria universidade e se tornando um dos órgãos de destaque da universidade. Hoje, destaca-se por incorporar um ambiente de rede complexo, heterogêneo e de alta tecnologia.

4.6 A redeUFSC

Com o objetivo de integrar todos os departamentos, cursos e laboratórios além dos órgãos administrativos e auxiliares – entre eles: biblioteca, hospital, museu, reitoria, etc., a redeUFSC foi, ao longo dos anos, crescendo e se desenvolvendo continuamente.

De um único computador, para um *mainframe* com terminais, posteriormente interligando em rede a milhares de micros. Inicialmente com tecnologia *ethernet*, depois *fast-ethernet*, migrando então para um *backbone* FDDI, que hoje se encontra sendo substituído pelo *backbone* ATM. Ver Figura 27.

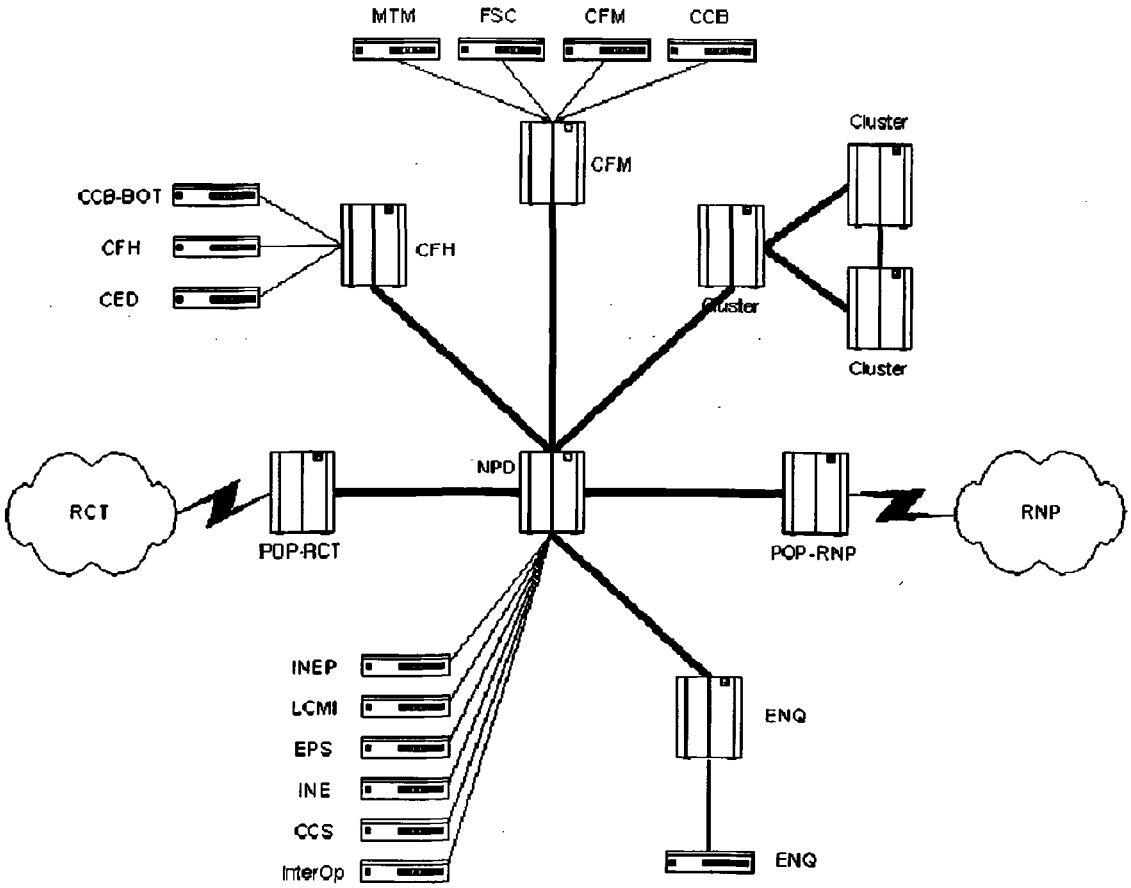

Figura 27: O *backbone* da redeUFSC.

A rede UFSC integra atualmente, computadores de diferentes plataformas (Sun, PC, RISC, Macintosh, LINUX, etc.), formando um ambiente composto por mais de 2.000 (dois mil) computadores.

4.7 Interconexões WAN da UFSC

A UFSC, através do seu NPD, vem se firmando como um dos grandes pontos de tráfego de rede do país. Inicialmente, por ser o ponto de presença da RNP no estado, centralizando o tráfego das instituições educacionais e de pesquisa.

Mas também por coordenar o projeto da Rede de Ciência e Tecnologia de Santa Catarina, que reúne todas as instituições de Ensino e Pesquisa do estado, num *backbone* modelo para o país.

E finalmente por coordenar o projeto da Rede Metropolitana de Alta Velocidade de Florianópolis, que é um ambiente de destaque, pela alta tecnologia empregada em seu projeto.

O fato da UFSC concentrar todas estas conexões (ver Figura 28), aliado ainda ao tamanho da sua redeUFSC, justifica a sua marcante representatividade no cenário nacional.

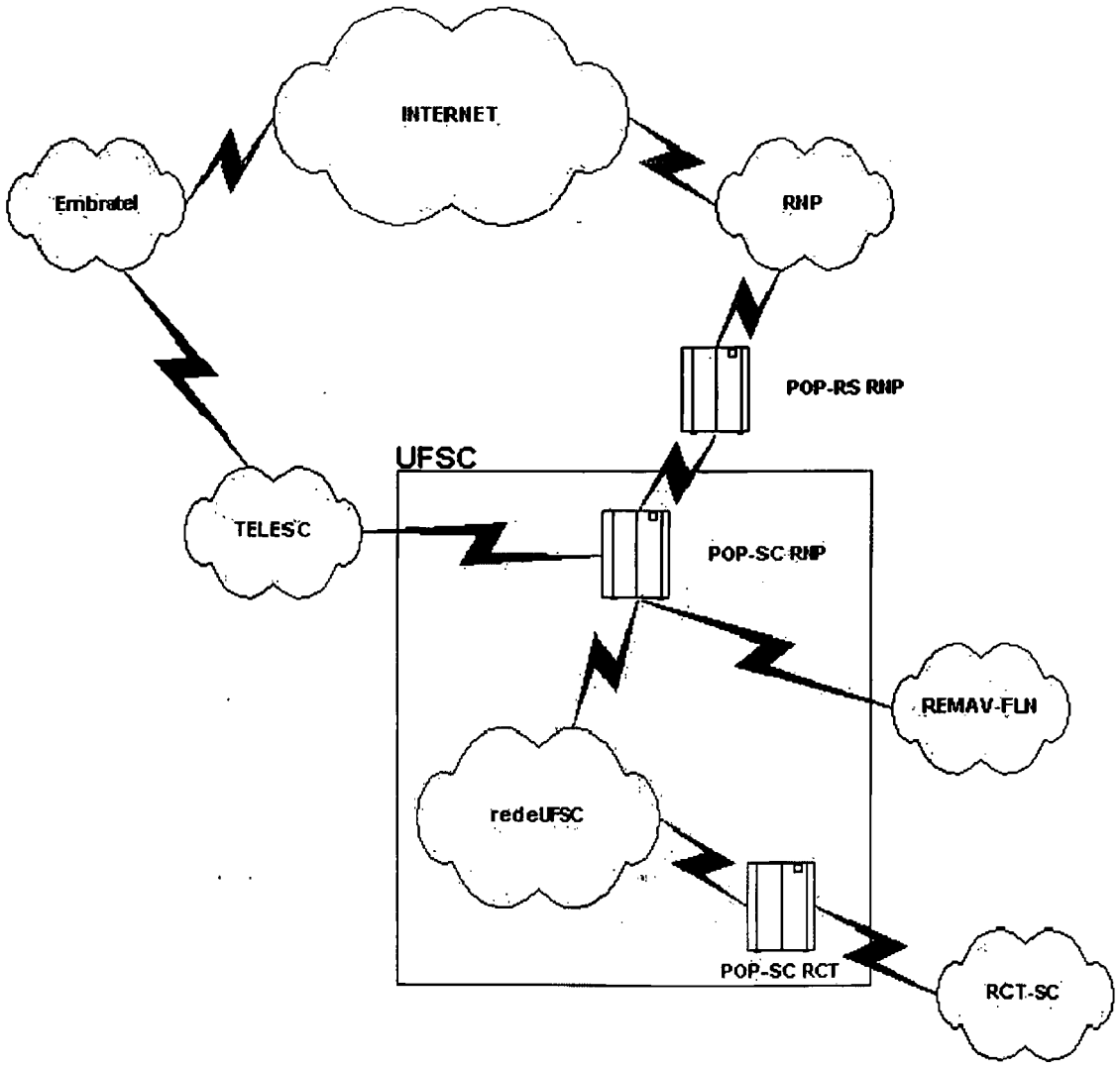

Figura 28: As conexões WAN da UFSC.

4.8 O ambiente de testes e experimentos

Objetivando testar a gerência e a monitoração de redes via *Web* com SNMP, foi necessário definir um ambiente para testes e experimentos. Ambiente este que serve como amostra da capacidade de monitoração e gerência, que uma ferramenta de gerência de redes *Web* pode apresentar.

O ambiente de testes está localizado dentro da redeUFSC, tendo como elemento principal um *switch* ATM IBM 8265. Este *switch* tem interfaces conectadas com os *switch's*: da redeUFSC; do CTC (Centro Tecnológico); do CFM (Centro de Ciências Físicas e Matemáticas); e 4 conexões com o NPD. Conforme apresenta a Figura 29.

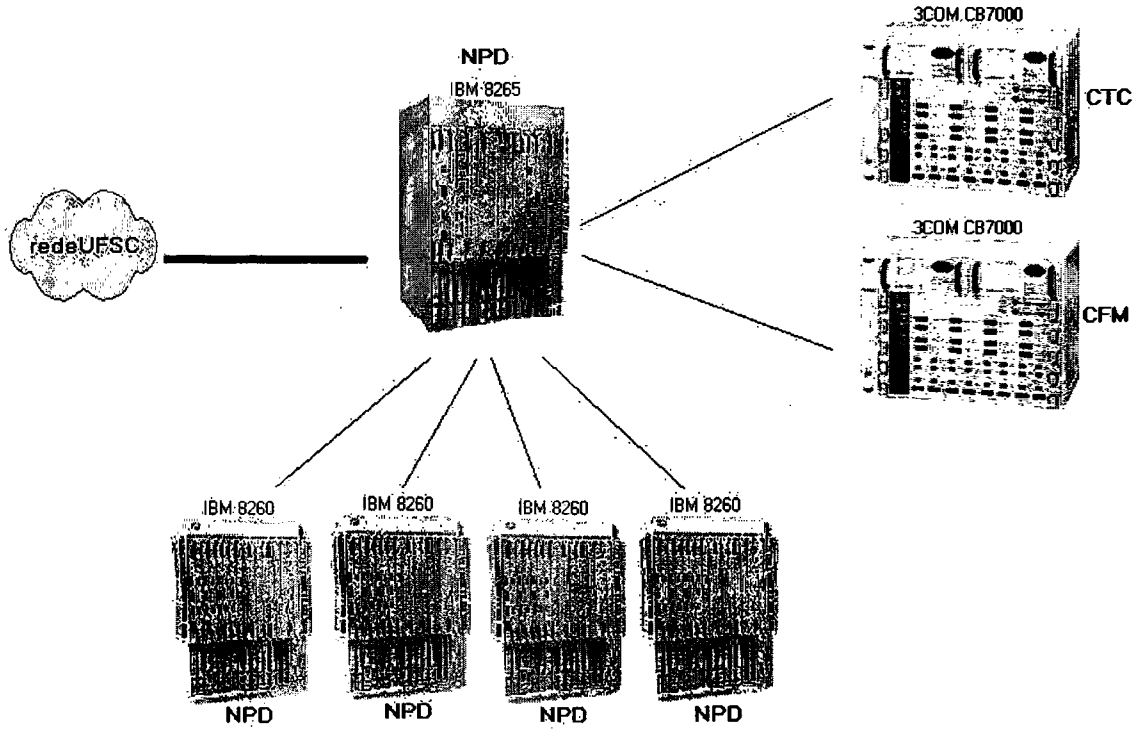

Figura 29: O ambiente de testes e experimentos.

O ambiente definido para experimentação apresenta uma boa estrutura para a realização de desenvolvimento e de testes, uma vez que as interfaces do *switch* central possuem diferentes características quanto à disponibilidade, demanda por tráfego e velocidade.

Quanto à disponibilidade, as interfaces com a redeUFSC, com o CTC e com o CFM, estão ativas todo o tempo. Quanto à velocidade, a interface com a redeUFSC trabalha com 622 Mbps, enquanto as demais interfaces trabalham com 155 Mbps. Quanto ao tráfego, existem diferentes demandas, que podem ser divididas em três grupos: a interface com a redeUFSC – de alta demanda, a interface com o CTC – de média demanda e as demais interfaces – de baixa demanda.

Quatro interfaces com o NPD estão invariavelmente disponíveis, o que pode ser considerado bom para a experiência de monitoração e gerência.

Nesse ambiente, a gerência se dá basicamente através do *switch* central, o IBM 8265 e das suas interfaces. A monitoração analisa diferentes tipos de informações, permitindo analisar o status do *switch* e visualizar o tráfego nas conexões.

Essa abordagem de gerência baseada num *switch* central, e permite a gerência *Web*, suas vantagens e desvantagens, podendo ser ampliada para gerenciar uma rede que possua vários *switch's* sem nenhum óbice.

Capítulo 05: SIGMA/WS – Ferramenta para a Gerência de redes ATM, via WWW, Java e SNMP

Capítulo 05: SIGMA/WS – Ferramenta para a Gerência de Redes ATM, via WWW, Java e SNMP

Uma ferramenta de gerência de redes baseada em um ambiente *Web*, com recursos para monitorar uma rede ATM através do SNMP, foi desenvolvida. Essa ferramenta é denominada SIGMA/WS – Ferramenta para a Gerência e a Monitoração de redes ATM via *Web* com SNMP.

A implementação da SIGMA/WS tem como objetivo examinar as possibilidades de monitoração e de gerência de uma rede ATM através do ambiente *Web*. O teste dessa ferramenta é realizado através de experimentos e monitorações.

Considerando que as redes ATM são compostas por *switch's* e pelos enlaces que os interconectam (ALLES, 1994); e que um único *switch* – com seus enlaces, pode representar o comportamento dos demais, definiu-se o ambiente de teste, restrito a um *switch* e às suas conexões. O desenvolvimento feito para um *switch* e as suas conexões pode ser replicado nos demais *switch's* da rede.

Visando aproveitar a característica de **navegabilidade hipertexto** do ambiente *Web*, adota-se a idéia de que a página inicial da ferramenta deve mostrar, numa única e ampla visão, todos os elementos que compõem o ambiente de teste.

Desenvolvendo-se primeiramente a *home-page* da ferramenta, a qual apresenta basicamente: os equipamentos que compõem o ambiente (e que são hiperligações para as páginas específicas dos equipamentos) e os enlaces que os interconectam (que também são hiperligações para as páginas específicas do enlace).

Cada elemento do ambiente monitorado – seja ele um equipamento gerenciável ou um enlace, possui uma hiperligação para uma outra página que contém maiores detalhes sobre ele. No Anexo A, encontra-se o código fonte, em HTML, da página principal da ferramenta, apresentada na Figura 30 a seguir.

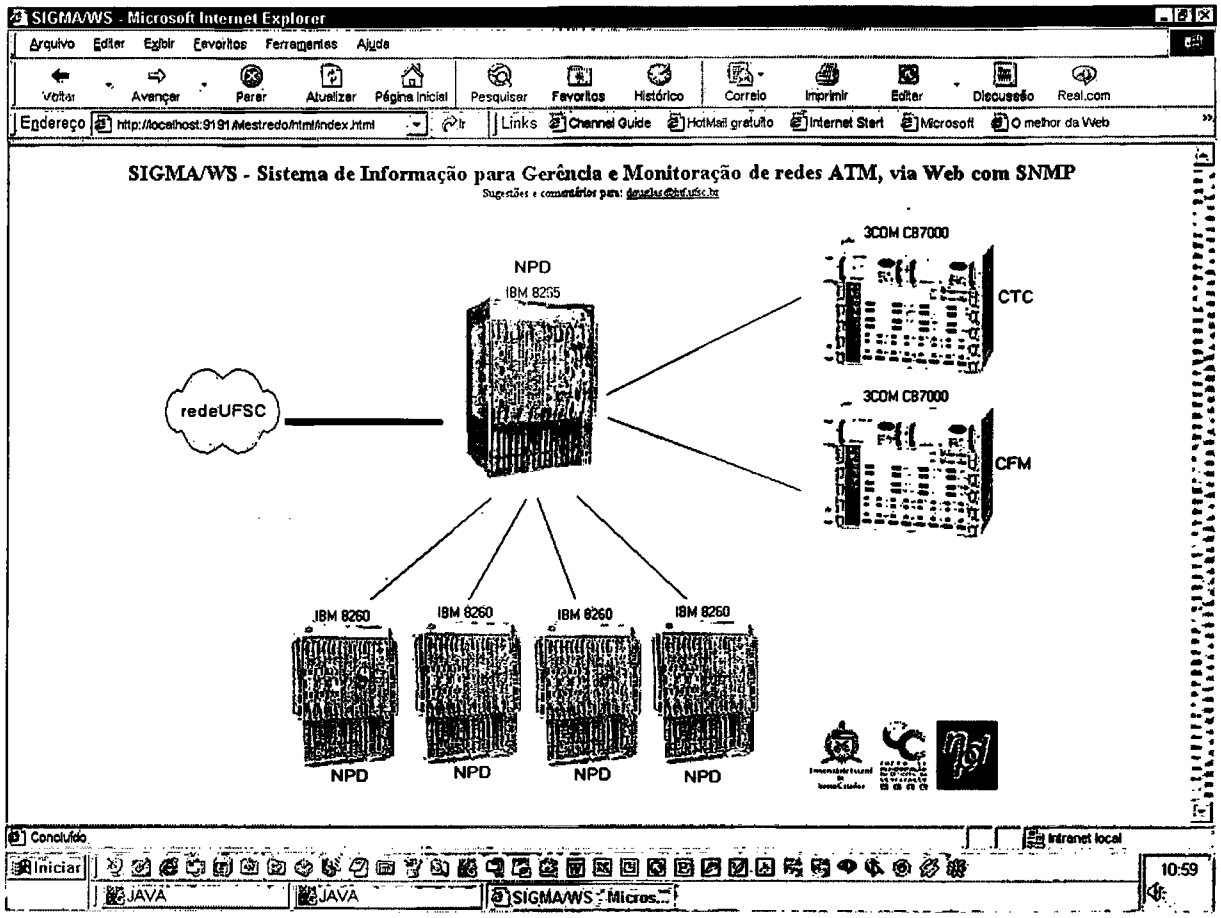

Figura 30: A *home-page* da ferramenta.

Conforme a Figura 30 apresenta, o elemento central do ambiente de teste é o *switch* IBM 8265 do NPD. Este *switch* possui enlaces que o conectam: à redeUFSC; ao CTC; ao CFM e quatro enlaces experimentais a outros *switch's* do NPD.

Considerando que as páginas desenvolvidas para a monitoração e gerência de um enlace são idênticas às páginas dos outros enlaces, optou-se por desenvolver as páginas das conexões com: a redeUFSC; o CTC; e o CFM, sem fazê-lo para os quatro enlaces do *switch* com o NPD. Pois, através da monitoração das três conexões – que possuem um volume maior de tráfego, pode-se examinar as possibilidades de uso da ferramenta.

Como as redes ATM são constituídas por *switch's* e pelos enlaces que os interconectam (ALLES, 1994), o desenvolvimento de um ambiente que se proponha a monitorá-lo fica simplificado. Explorando essa simplificação, pode-se desenvolver

apenas dois tipos de páginas: as páginas sobre os equipamentos gerenciados e as páginas sobre os enlaces que os interconectam.

A seguir, a ferramenta é apresentada em duas seções: 5.1, com as informações do *switch* (dividida em informações gerais, informações sobre o estado das conexões e informações sobre o tráfego nas conexões) e 5.2, com as conexões do *switch* (dividida em o volume da conexão; a vazão da conexão e a qualidade de serviço da conexão).

5.1 As Páginas com Informações do *Switch*

A página que apresenta as informações do *switch* visa detalhar um conjunto significativo de dados. O volume de informações a ser disponibilizado é tal, que elas foram divididas em categorias e apresentadas em páginas distintas às quais se tem acesso a partir da página principal.

Uma vez que o ambiente de teste, no qual foi baseada a implementação, tem como elemento central o *switch* IBM 8265, as páginas com as informações sobre esse *switch* foram desenvolvidas e testadas nesse switch. No Anexo B, encontra-se o código fonte em HTML da página sobre o *switch* IBM 8265, apresentada na Figura 31.

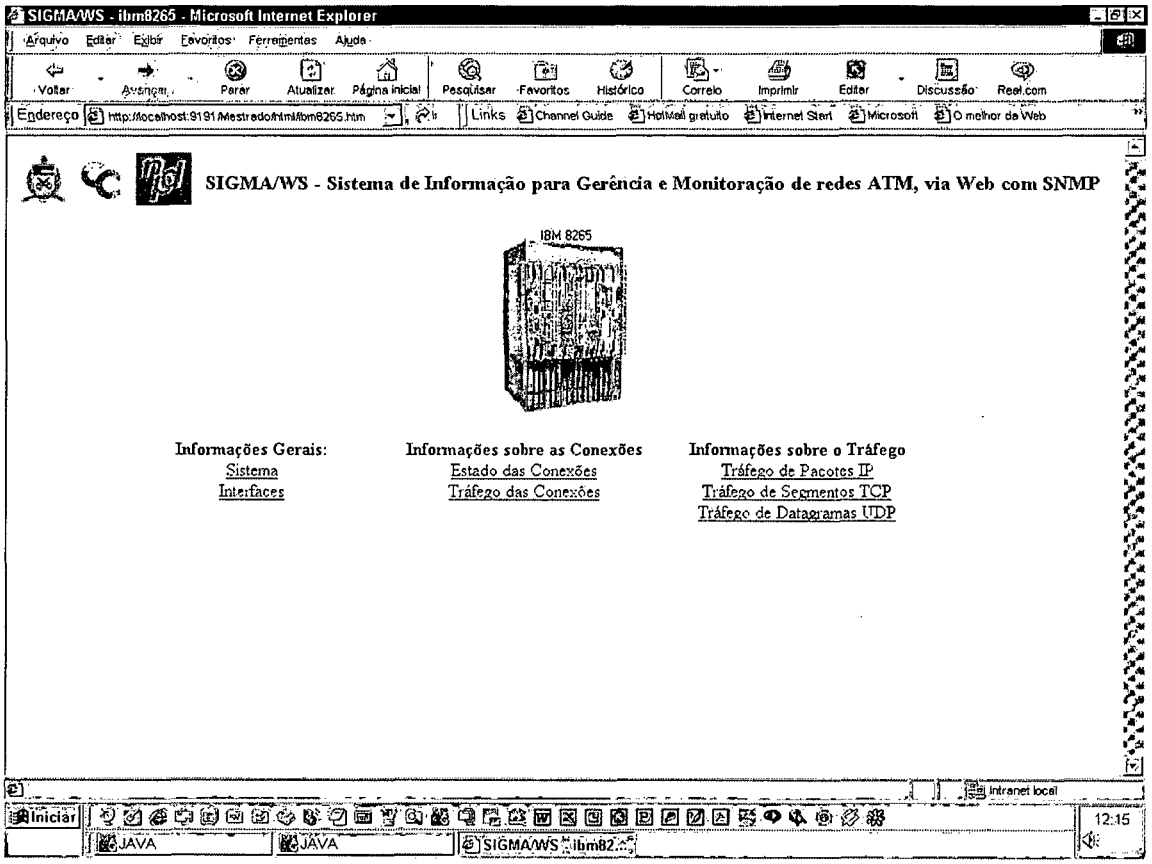

Figura 31: A página de informações sobre o *switch* IBM 8265.

Conforme apresentado na Figura 31, a página de informações do *switch* IBM 8265 traz sua imagem e algumas hiperligações que oferecem acesso a outras páginas, as quais contêm informações mais específicas. Estas hiperligações estão divididas em três categorias distintas, a saber, informações gerais, informações sobre as conexões e informações sobre tráfego.

As informações gerais do *switch* são apresentadas em duas páginas, a de sistemas e a de interfaces.

As informações sobre as conexões são apresentadas em duas páginas: o estado das conexões e o tráfego das conexões. As informações sobre o tráfego são apresentadas em três páginas o tráfego de pacotes IP, o tráfego de segmentos TCP e o tráfego de datagramas UDP. A seguir são descritas essas páginas.

5.1.1 As páginas com Informações Gerais sobre o *switch*

As páginas de informações gerais destinam-se a adequada manutenção e operação da rede. Elas são divididas em duas páginas: informações sobre sistema; e informações sobre as interfaces. Na primeira página, são apresentadas informações genéricas (relacionadas ao equipamento como um todo). Na segunda página são apresentadas informações mais específicas (sobre o equipamento e suas interfaces).

A página de Informações sobre o Sistema

As informações apresentadas na página de Informações sobre o Sistema têm como principal objetivo prover facilidades para a equipe de manutenção e operação de rede. Quando for desejável prestar um serviço de manutenção e operação em uma rede, independentemente do tamanho da mesma, deve-se manter um cadastro de informações organizado e atualizado.

As informações necessárias para a adequada manutenção e operação da rede, podem ser agrupadas em informações sobre os computadores e os periféricos (equipamento, marca, modelo, processador, memória, disco rígido, sistema operacional, etc.), informações sobre os programas (que programa é usado, número de licença, se é cliente ou servidor, de quem é cliente ou servidor, etc.) e informações sobre a rede (os elementos de rede, as conexões e as suas características).

A maioria das informações relacionadas à rede, e necessárias para a sua manutenção, pode ser armazenada e mantida atualizada nos equipamentos gerenciáveis. Os equipamentos gerenciáveis permitem que se possa manter as informações úteis armazenadas neles. Um ambiente *Web* que apresente a topologia da rede, sensível ao contexto, permite que se possa analisar a rede e os seus equipamentos.

Cabe lembrar que muitos dos serviços de manutenção e operação de rede normalmente são feitos em locais distantes, e uma ferramenta *Web* está acessível de qualquer local, dentro e fora da rede.

Dentre os serviços que podem ser facilitados com o armazenamento e a busca de informações diretamente nos elementos gerenciáveis, através da *Web*, podem ser citados: verificar a versão de software do equipamento; determinar a sua localização na organização, identificar o profissional responsável e como localizá-lo; identificar o tipo de serviço prestado pelo equipamento; verificar há quanto tempo o sistema está ativo; ver a que domínio ele pertence; e obter a identificação do seu fabricante.

No Anexo C, encontra-se o código fonte em HTML da página com informações sobre o sistema, e no Anexo C1, está o código fonte em Java do applet utilizado na referida página, a qual é apresentada na Figura 32 abaixo.

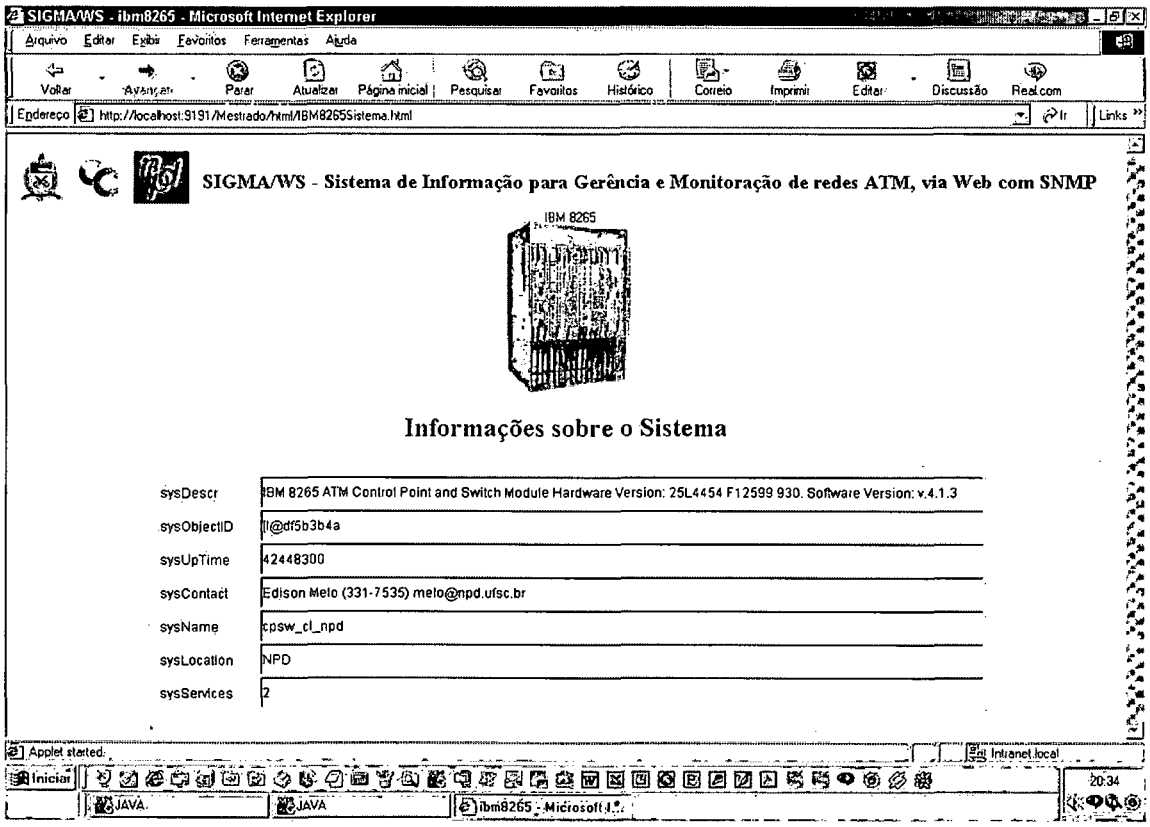

Figura 32: A página de informações sobre o sistema.

Conforme mostra a Figura 32, a página de informações sobre o sistema apresenta informações do *switch* e de seu funcionamento, mais precisamente: a descrição; o nome; o identificador do objeto; o tempo de atividade; o contato; a localização e o tipo de serviço.

A seguir são descritas as informações contidas na página de informações sobre o sistema. Descrevem-se também a variável da MIB correspondente e qual a sua função. Todas as variáveis utilizadas nesta página são obtidas da MIB II (RFC-1213).

sysDescr é a variável da MIB-II normalmente utilizada para armazenar um texto que identifique o nome e a versão completa do hardware e do software (sistema operacional e de rede).

sysObjectID é a variável da MIB-II utilizada para armazenar um texto que identifique o equipamento. Esta identificação é pré-determinada pelo fabricante, através de sua localização na MIB proprietária do mesmo.

sysUpTime é a variável da MIB-II utilizada para armazenar o tempo de atividade do sistema de gerência, ou seja, o tempo durante o qual o sistema de gerência permanece ativo. O tempo é apresentado em centésimos de segundo. Ressalta-se que o sistema de gerência de rede pode ser reinicializado, sem que o elemento de rede precise de reinicialização.

sysContact é a variável da MIB-II utilizada para armazenar um texto que deve identificar a pessoa com a qual contatar com relação a um elemento gerenciável, junto com a informação de como fazê-lo.

SysName é uma variável da MIB-II utilizada para armazenar um texto que determine o nome dado para um nodo gerenciável, com uma conotação administrativa, e que, por convenção, tem sido utilizado para armazenar a especificação completa do nome de domínio do nodo.

sysLocation é uma variável da MIB-II que serve para armazenar a informação da localização física do nodo, determinando o local no qual o equipamento encontra-se fisicamente instalado, por exemplo: CTC, 1° andar, sala 15.

sysServices é uma variável da MIB II que é empregada para armazenar o tipo de serviço que o elemento de rede primariamente oferece, podendo ser um dos seguintes: 1 físico (repetidores); 2 enlace de dados/sub-redes (pontes); 3 Internet (IP *gateways*); 4 ponto-a-ponto (IP *host's*); e 7 aplicações (envio de e-mail).

A página de Informações sobre as Interfaces Físicas

A página de informações sobre as interfaces físicas visa reunir informações que possam ser úteis ao suporte de rede, no que tange ao estado operacional de cada uma das interfaces do equipamento.

Os dados contidos na página de informações sobre as interfaces físicas permitem a quem exerce a atividade de suporte de rede descobrir remotamente, ou seja de sua própria mesa, se um defeito ocorre exclusivamente em uma interface física, ou em todas as interfaces de um *slot* (ou seja, em uma placa de interfaces), ou, ainda, se em todas as interfaces do *switch* (ou seja, em todo o *switch*).

Na página de informações sobre as interfaces, procura-se apresentar, dentre as informações relativas às interfaces de conexões disponíveis nas MIB's, aquelas que oferecem maiores facilidades para o suporte da rede. Para obter uso mais geral, são empregadas variáveis da MIB-II (RFC-1213), e variáveis da MIB proprietária IBM.

No Anexo D, encontra-se o código fonte, em HTML, da página com informações sobre as interfaces físicas, e no Anexo D1, está o código fonte, em Java, do applet utilizado na referida página, a qual é apresentada na Figura 33 a seguir.

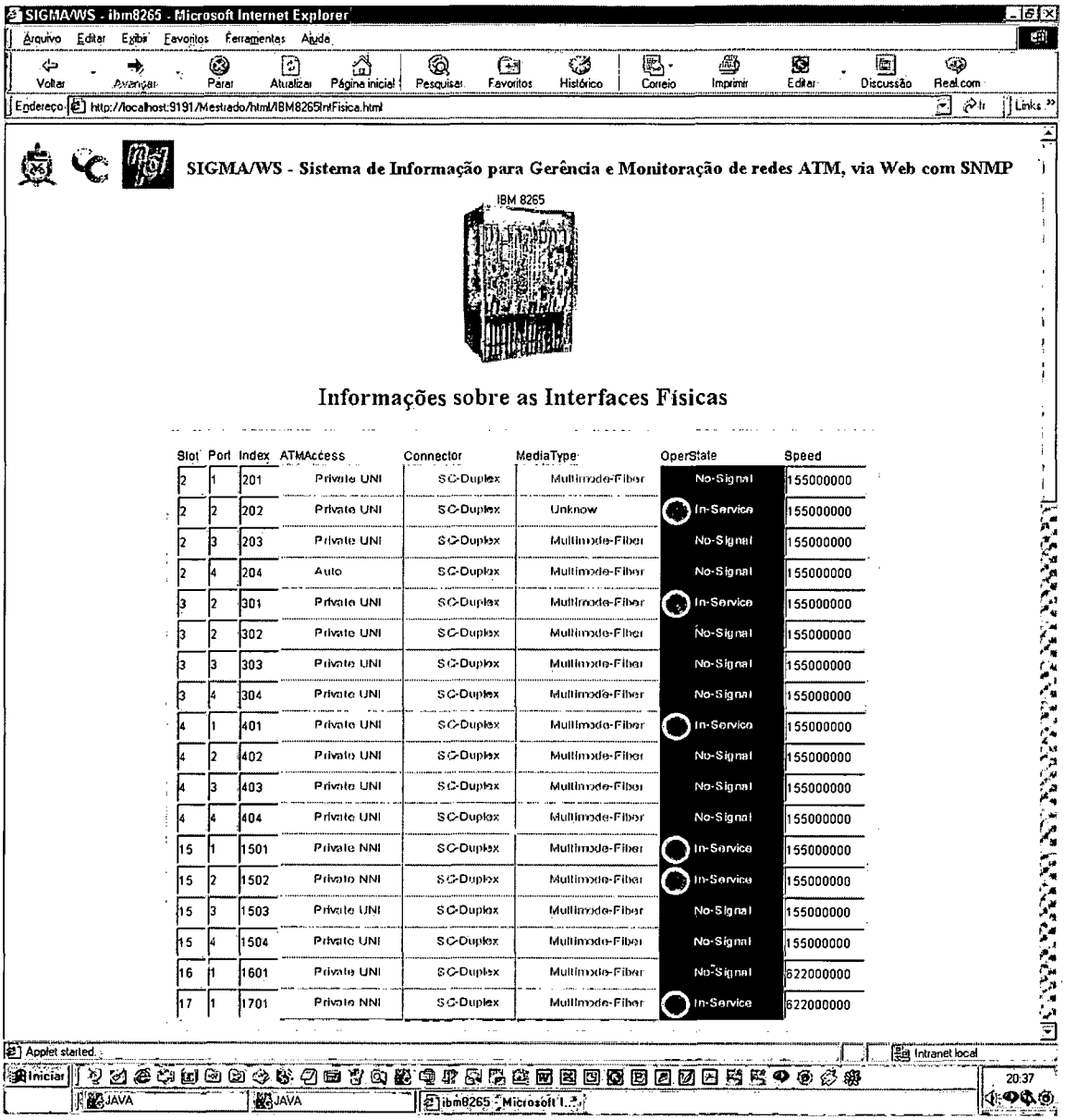

Figura 33: A página de informações sobre as interfaces físicas.

A seguir são descritas as variáveis de MIB apresentadas na Figura 33. Essas variáveis são utilizadas para a obtenção dos dados contidas na página de informações sobre as interfaces físicas, procurando indicar sua função e de qual MIB é obtida.

A interface física de um *switch's* é normalmente um conector embutido em uma placa que é adaptada em *slot* no *switch*. As placas de interfaces físicas podem apresentar mais do que somente uma interface física. Desse modo, para indicar fisicamente e univocamente uma interface física no *switch* faz-se necessário informar o número do

slot ao qual a placa de interfaces físicas está conectada e o número da interface física na placa.

interfaceSlot é a variável da MIB-II (RFC-1213) disponível para armazenar o número do *slot* no qual a placa de interfaces físicas está inserida. Um *switch* vem com um número limitado de slot's que são utilizados à medida que é necessário.

interfacePort é a variável da MIB-II (RFC-1213) que pode ser utilizada para armazenar o número da porta (ou seja, do conector), na placa de interfaces físicas.

interfaceIndex é a variável da MIB-II (RFC-1213) usada para armazenar um índice de interfaces, ou seja, um número que indica univocamente a interface física. Observando a Figura 33, percebe-se que o índice de interfaces ***(interfaceIndex)*** é uma junção do número do slot ***(interfaceSlot)*** com o número da porta ***(interfacePort)***.

Considerando que outro equipamento – principalmente de outra marca, poderia não seguir esta regra, resolveu-se incluir a informação do índice de interfaces na página de informações sobre as interfaces físicas.

InterfaceATMaccess é a variável da MIB-II (RFC-1213) que armazena um valor que indica o tipo de acesso ATM oferecido na interface física. Esta informação tem relação direta com a estrutura de acesso da mesma, e com o tipo de equipamento que está sendo conectado. O valor apresentado pode ser: 1 (*Unknown*); 2 (*Private UNI*); 3 (*Private NNI*); 4 (*Public UNI*); 5 (GSMP - *Generic Switch Management Protocol*); 7 (*Void*); 8 (*Auto*); e 9 (*Snoop*).

interfaceConnector é a variável da MIB proprietária da IBM que armazena um valor, o qual indica o tipo de conector utilizado na interface, a saber: 1 (*Unknown*); 2 (Internal); 3 (Mic); 4 (SC-Duplex); 5 (Monomode); 6 (DB-9); 7 (RJ45); 8 (BNC); e 9 (DB-15).

interfaceMediaType é a variável da MIB proprietária da IBM que armazena um valor que indica o tipo de mídia utilizado nesta interface, podendo ser: 1 (*Unknown*); 2 (Fibra Mono-modo); 3 (Fibra Multi-modo); 4 (Par trançado); 5 (UTP), 6 (STP); 7 (cabo coaxial); 8 (Placa mãe); 9 (Fibra de longo alcance) e 10 (Fibra não especificada).

O tipo de mídia e o tipo de conector utilizado na interface, são informações importantes para quem vai fazer a manutenção de um enlace de rede, porque eventualmente, o defeito apresentado pode estar tanto no cabo quanto no conector. A possibilidade de identificar remotamente qual o tipo do cabo ou da mídia elimina a necessidade de verificação *in loco*.

ifSpeed é a variável da MIB-II (RFC-1213) utilizada para armazenar a largura de banda da interface. Ela é apresentada em bits por segundo. A informação que apresenta a velocidade do tráfego também contribui para uma análise preliminar, que venha a ser feita remotamente, principalmente se a interface permitir variação. Para interfaces nas quais não há variação na largura de banda, porém, ou para aquelas onde a estimativa não pode ser feita, a variável ***ifSpeed*** contém a largura de banda nominal.

ifOperStatus é a variável da MIB-II (RFC-1213) que armazena o valor que indica o estado operacional da interface física, a saber: 1 (up); 2 (down); e 3 (testing). Estes três valores - oferecidos pela MIB-II (RFC-1213), não são específicos, se levarmos em consideração a quantidade de problemas e/ou situações diferentes que podem ser encontradas numa interface.

interfaceOperState é a variável da MIB proprietária da IBM utilizada para armazenar um valor que indica o estado operacional da interface – a exemplo da variável ***ifOperStatus*** da MIB-II (RFC-1213), porém oferece um grau maior de esclarecimento para o usuário. Podendo apresentar, assim, informações mais precisas, para o adequado suporte da rede.

Os valores que podem ser encontrados na variável ***interfaceOperState*** são os seguintes: 1 (unknown); 2 (disabled-no-signal); 3 (disabled-idle); 4 (no-signal); 5 (idle); 6 (in-service); 7 (in-service-no-address-registration); 8 (failing-internal); 9 (misConfigured); 10 (wrongNetworkPrefix); 11 (wrongNodeNumber); 12 (disabled-failing); 13 (failing-line); 14 (wrap-no-signal); 15 (wrap-idle); 16 (wrap-failing-internal); 17 (wrap-failing-line); 18 (idle-no-bandwidth); 19 (idle-internal-error); 20 (disabled-no-bandwidth); 21 (wrap-far-end-no-signal); 22 (wrap-far-end-idle); 23 (wrap-far-end-failing); 24 (wrap-far-end-failing-line); 25 (insufficient-connection-handles); 26 (invalid-remote-vpi-vci-range); 27 (control-vpi-already-used); e 28 (missing-signaling-version).

5.1.2 As páginas com Informações sobre o Estado das Conexões

Através das páginas que possuem informações sobre o estado das conexões do *switch,* o usuário do SIGMA/WS pode ter uma visão ampla, que contemple todas as conexões do *switch* ao mesmo tempo, de maneira a permitir uma completa percepção do estado das conexões.

Para dar esta visão, de completa percepção do estado do *switch* e de seus enlaces, duas páginas foram desenvolvidas: a página de informações sobre o estado das conexões e a página de informações sobre o tráfego nas conexões. A seguir, estas páginas são descritas.

A página de Informações sobre o Estado das Conexões

Para facilitar a compreensão do estado operacional dos enlaces do *switch*, a página de informações sobre as conexões apresenta a imagem do *switch* no centro, cercada por suas conexões, sobre cada conexão tem-se o estado operacional de cada uma das conexões.

No Anexo E, encontra-se o código fonte, em HTML, da página com informações sobre o estado das conexões, e no Anexo E1, está o código fonte, em Java, do applet utilizado na referida página, a qual é apresentada na Figura 34 abaixo.

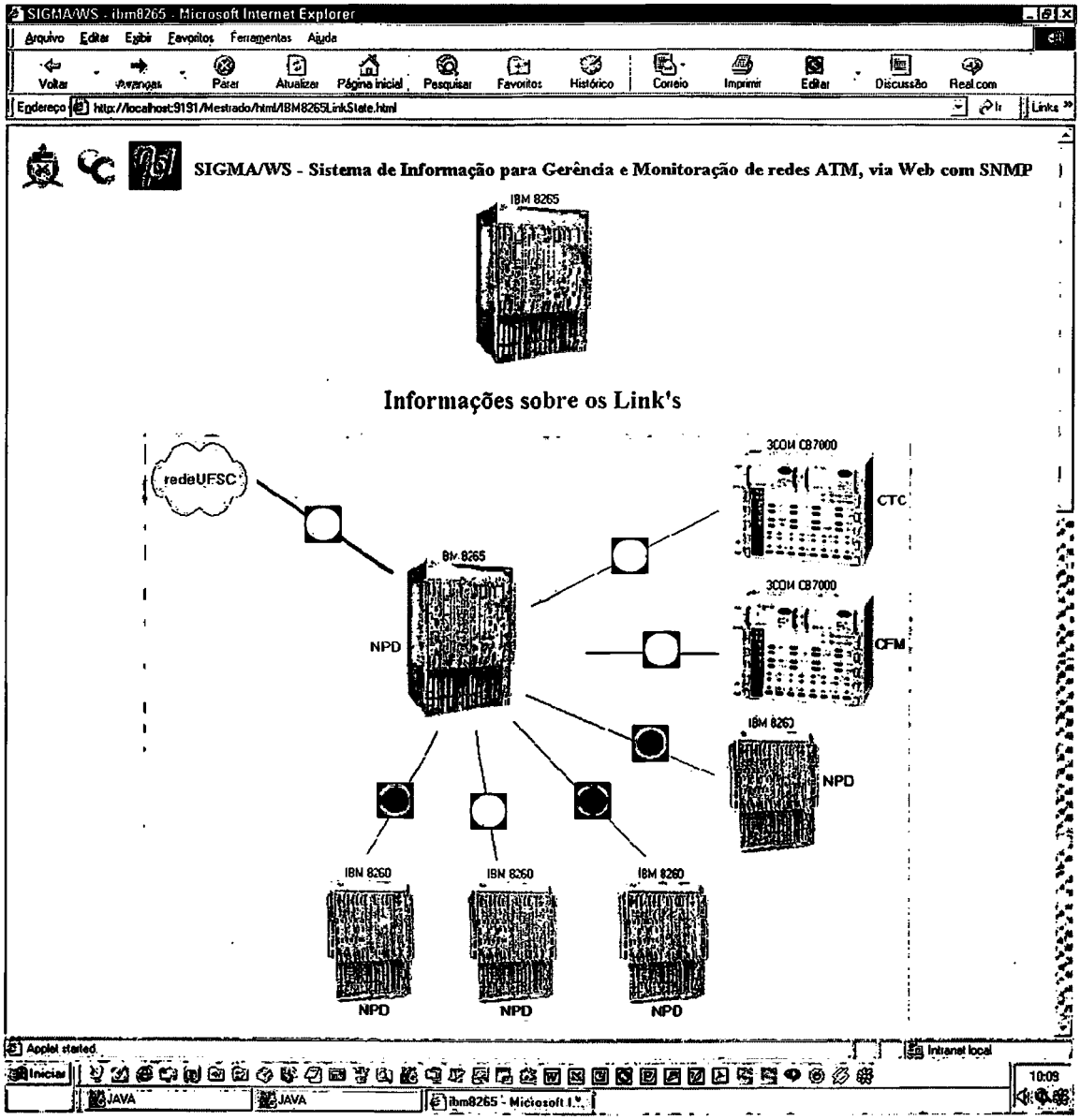

Figura 34: A página de informações sobre o estado das conexões.

Como a idéia desta página é permitir ao usuário do SIGMA/WS uma rápida noção do estado das conexões, julgou-se suficiente a apresentar a informação do estado operacional provida pela MIB-II (RFC-1213), através da variável ***ifOperStatus***.

A simplicidade da informação apresentada na página de informações sobre o estado das conexões – somente a variável ***ifOperStatus***, justifica-se, em caso de necessidade, o usuário pode obter informações mais detalhadas pela página de informações sobre as interfaces físicas.

A variável ***ifOperStatus*** pode assumir os seguintes valores: 1 (*Up*); 2 (*Down*); ou 3 (*Testing*). Para facilitar a visualização, o valor é representado por cores – conforme pode ser visto na Figura 34, a saber: *Up* (verde); *Down* (vermelho); e *Testing* (Azul).

A página sobre o Tráfego nas Conexões

A página que apresenta as informações sobre o tráfego nas conexões tem por objetivo dar uma visão global do nodo de rede gerenciado, permitindo comparar uma conexão em relação às outras para analisar o nodo de rede.

O tráfego sobre um enlace é uma informação importante, que deve ser analisada detalhadamente, para permitir avaliar o estado desse tráfego. Apesar da ampla visão que todos os enlaces juntos oferecem, através dessa página é possível determinar, preliminarmente, se o tráfego está balanceado ou não.

Com algum conhecimento no que diz respeito às aplicações utilizadas, é possível identificar a origem e o destino do fluxo de dados, o que permite avaliar melhor os volumes de tráfego nas interfaces. As informações apresentadas na página de informações sobre o tráfego dos enlaces são obtidas das variáveis ***ifInOctets*** e ***ifOutOctets***, através da MIB-II (RFC-1213).

No Anexo F, encontra-se o código fonte, em HTML, da página com informações sobre o tráfego nas conexões, e no Anexo F1, está o código fonte, em Java, do applet utilizado na referida página, a qual é apresentada na Figura 35, a seguir.

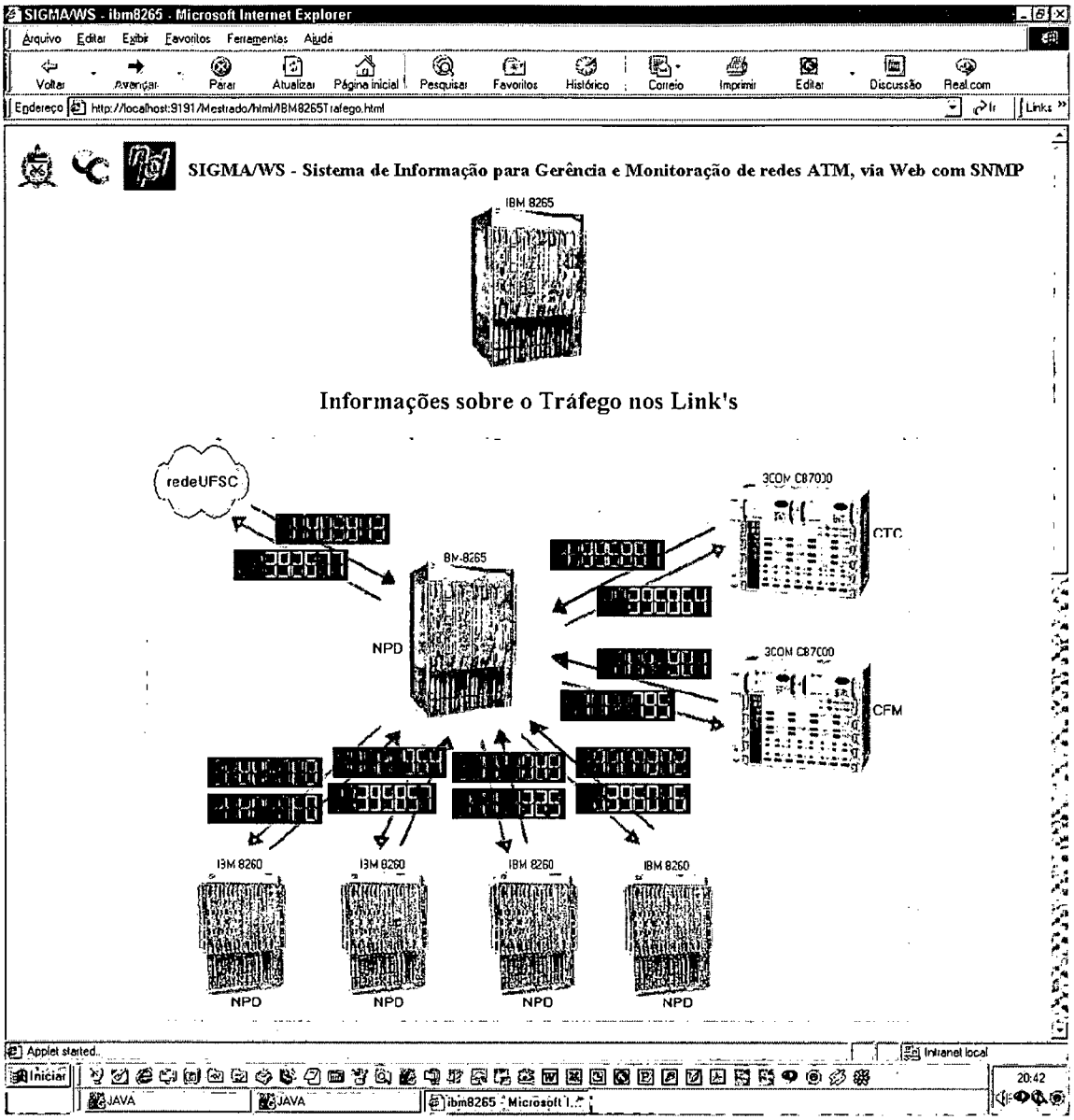

Figura 35: Página de informações sobre o tráfego nas conexões.

Conforme apresenta a Figura 35, a página de informações sobre o tráfego nas conexões oferece uma visão global do *switch*, de suas conexões, e do volume de tráfego que flui – entrando e saindo de suas conexões. Estas informações permitem que se possa observar o tráfego que passa pelo *switch*, verificando possíveis gargalos ou analisando o tráfego gerado por alguma aplicação.

5.1.3 Páginas com Informações sobre o Tráfego nas Conexões

As páginas de informação sobre tráfego nas conexões apresentam o volume de dados enviados e recebidos. As informações do tráfego nas conexões foram consideradas no contexto das "informações do *switch*" para permitir uma análise do tráfego do *switch*. Para tanto, foram desenvolvidas páginas para monitorar o volume de tráfego de pacotes IP; segmentos TCP; e datagramas UDP.

O uso dessas páginas é feito quando há necessidade de medir o volume de transmissões, de acordo com o tipo de tráfego. Além disso, podem ser empregadas para determinar se um eventual congestionamento é causado por um tipo específico de tráfego.

A seguir, são descritas as três páginas desenvolvidas para mostrar o volume do tráfego nas conexões, as variáveis utilizadas em cada uma delas. Considerando que as três páginas (de pacotes IP; de segmentos TCP; e de datagramas UDP), basearam-se no volume de dados enviados e recebidos, optou-se por um modelo único no desenvolvimento dessas páginas.

Página com Informações sobre o Tráfego de Pacotes IP

A página de informações sobre o tráfego de pacotes IP apresenta ao usuário o volume de pacotes IP transmitidos pelo *switch*. O volume é visto de duas maneiras: o volume de pacotes enviados; e o volume de pacotes recebidos.

A informação da quantidade de pacotes IP enviados e recebidos é obtida através das variáveis MIB-II (RFC-1213) ***ipInReceives*** e ***ipOutRequest***. Os dados obtidos são apresentados em dois gráficos distintos.

No Anexo G, encontra-se o código fonte, em HTML da página com informações sobre o tráfego de pacotes IP, e no Anexo G1, está o código fonte, em Java, do applet utilizado na referida página, a qual é apresentada na Figura 36, abaixo.

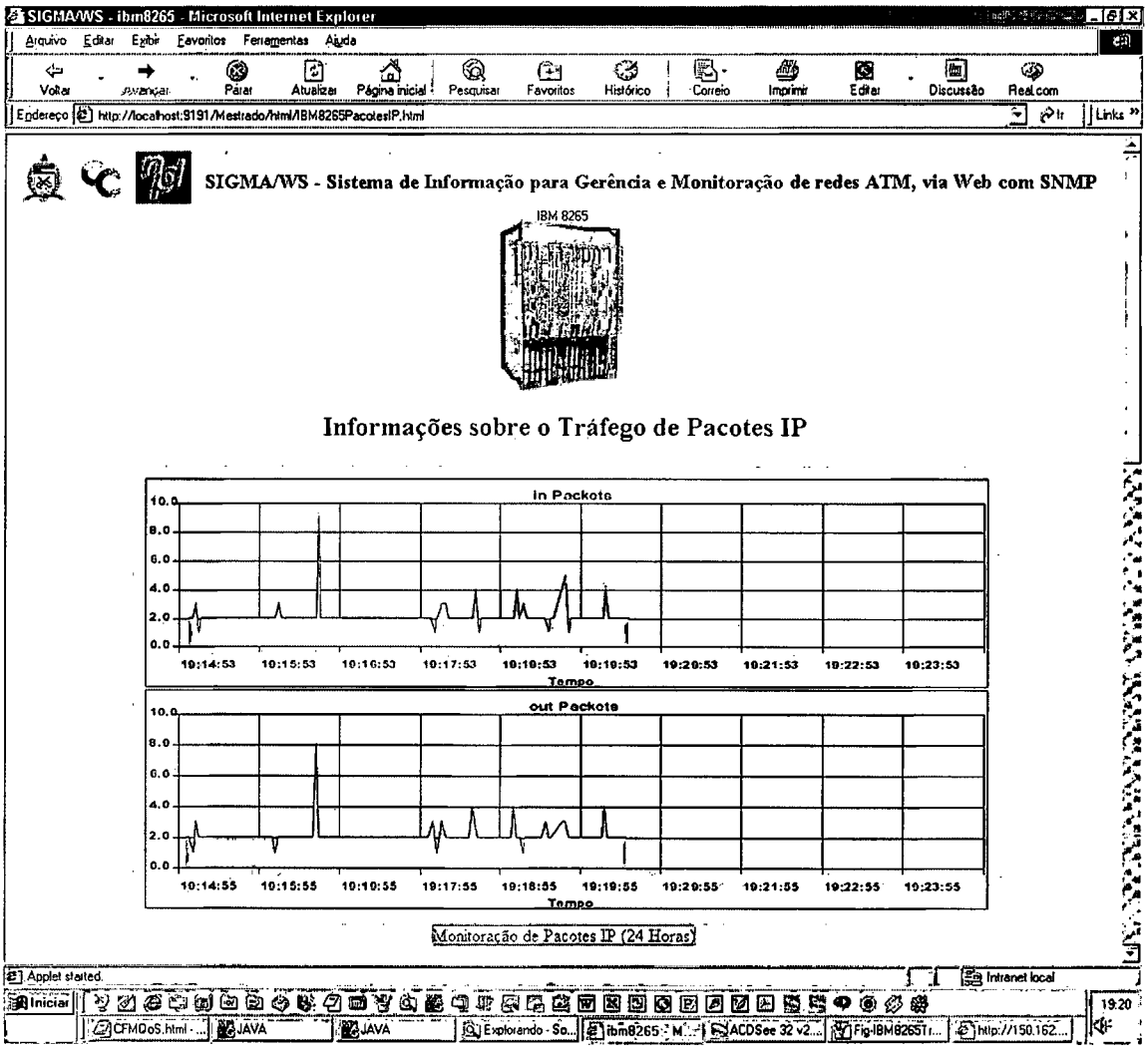

Figura 36: Página de informações sobre o tráfego de pacotes IP.

Conforme apresenta a Figura 36 os gráficos demonstram o tráfego de 10 (dez) minutos. Se o usuário quiser analisar o tráfego de pacotes IP por um período de tempo maior, recomenda-se o uso da hiperligação "Monitoração de Pacotes IP (24 horas)", no final da página.

Essa hiperligação executa a abertura de uma nova janela que apresenta gráficos semelhantes, baseados nas mesmas variáveis, que permitem porém, a visualização dos dados de um dia inteiro (24 horas).

No Anexo H, encontra-se o código fonte, em HTML da página com informações sobre o tráfego de pacotes IP (24 h), e no Anexo H1, está o código fonte, em Java, do applet utilizado na referida página.

A página com Informações sobre o Tráfego de segmentos TCP

A página de informações sobre o tráfego atuante sobre segmentos TCP apresenta ao usuário o volume de segmentos TCP transmitidos pelo *switch*. O volume é apresentado de duas maneiras: o volume de segmentos enviados; e o volume de segmentos recebidos.

A informação da quantidade de segmentos TCP enviados e recebidos é obtida através das variáveis MIB-II (RFC-1213) ***tcpInSegs*** e ***tcpOutSegs***. Os dados obtidos são apresentados em dois gráficos distintos.

No Anexo I, encontra-se o código fonte, em HTML, da página com informações sobre o tráfego de segmentos TCP, e no Anexo I1, está o código fonte, em Java, do applet utilizado na referida página, a qual é apresentada na Figura 37, a seguir.

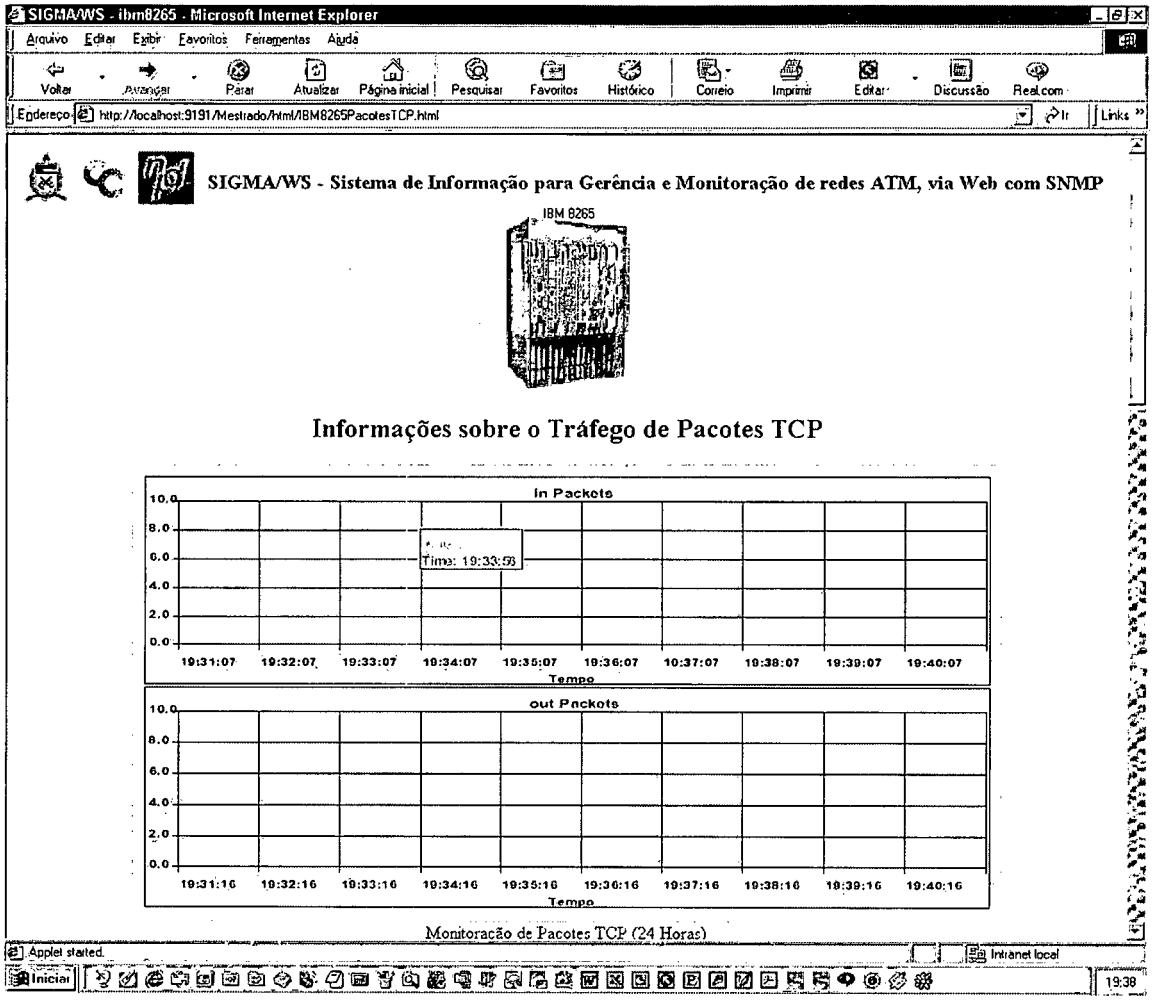

Figura 37: Página de informações sobre o tráfego de segmentos TCP.

Conforme apresenta a Figura 37, os gráficos demonstram o tráfego de 10 (dez) minutos. Se o usuário quer analisar o tráfego de segmentos TCP por um período de tempo maior, recomenda-se o uso da hiperligação "Monitoração de Pacotes TCP (24 horas)", no final da página.

Essa hiperligação executa a abertura de uma nova janela, que apresenta gráficos semelhantes, baseados nas mesmas variáveis, que permitem a visualização dos dados de um dia inteiro (24 horas).

No Anexo J, encontra-se o código fonte, em HTML, da página com informações sobre o tráfego de segmentos TCP (24 h), e no Anexo J1, está o código fonte, em Java, do applet utilizado na referida página.

Página com Informações sobre os Datagramas UDP

A página de informações sobre o tráfego de datagramas UDP apresenta ao usuário o volume de datagramas UDP transmitidos pelo *switch*. O volume é apresentado de duas maneiras: o volume de datagramas enviados; e o volume de datagramas recebidos.

A informação da quantidade de datagramas UDP enviados e recebidos é obtida através das variáveis MIB-II (RFC-1213) ***udpInDatagrams*** e ***udpOutDatagrams***. Os dados obtidos são apresentados em dois gráficos distintos.

No Anexo K, encontra-se o código fonte, em HTML, da página com informações sobre o tráfego de datagramas UDP, e no Anexo K1, está o código fonte, em Java, do applet utilizado na referida página, a qual é apresentada na Figura 38, a seguir.

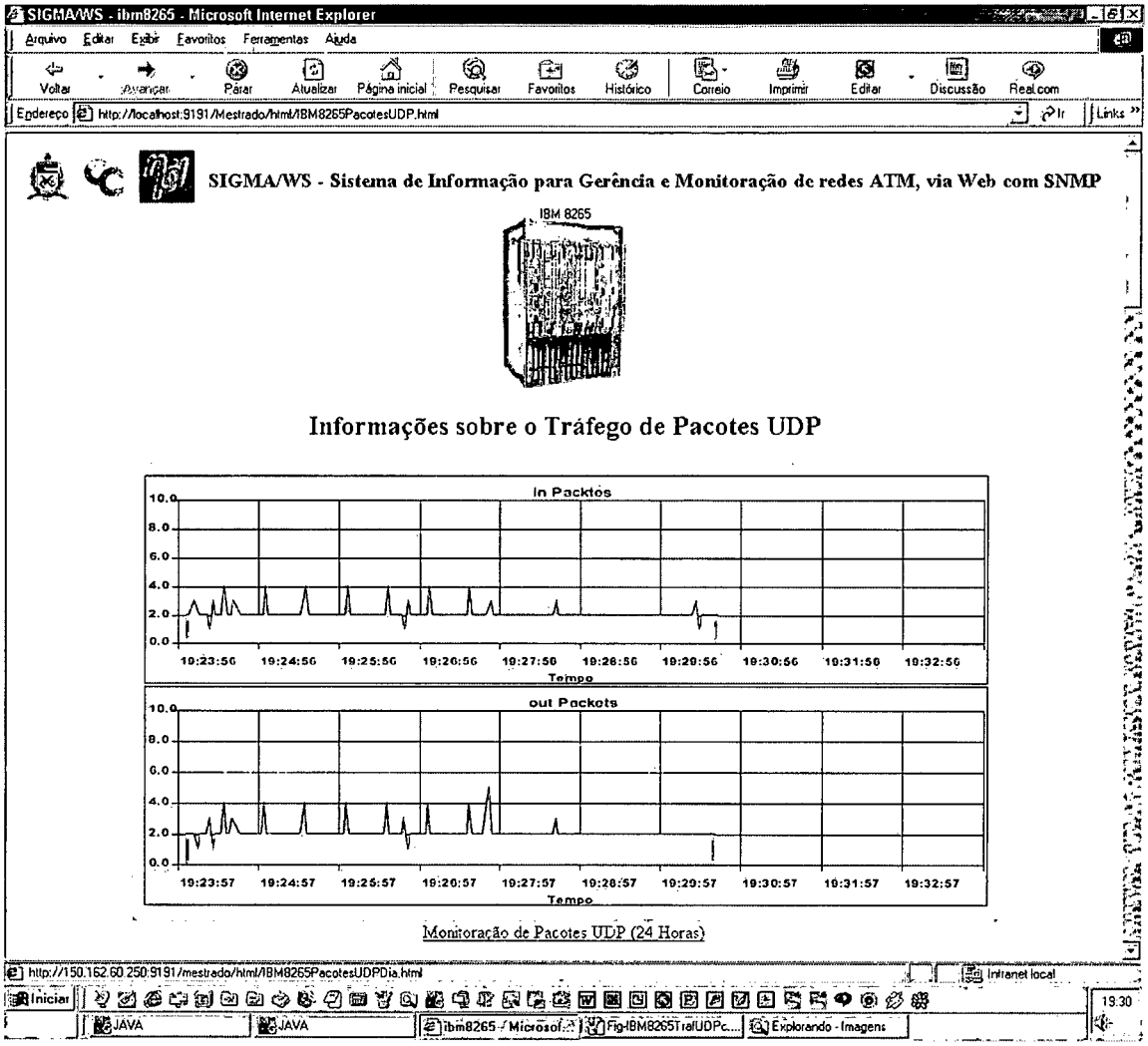

Figura 38: Página de informações sobre o tráfego de datagramas UDP.

Conforme apresenta a Figura 38, os gráficos demonstram o tráfego de 10 (dez) minutos. Se o usuário quiser analisar o tráfego de datagramas UDP por um período de tempo maior, recomenda-se o uso da hiperligação "Monitoração de Pacotes UDP (24 horas)", no final da página.

Essa hiperligação executa a abertura de uma nova janela, que apresenta gráficos semelhantes, baseados nas mesmas variáveis, que permitem a visualização dos dados de um dia inteiro (24 horas).

No Anexo L, encontra-se o código fonte, em HTML, da página com informações sobre o tráfego de datagramas UDP, e no Anexo L1, está o código fonte, em Java, do applet utilizado na referida página.

5.2 Páginas sobre as Conexões do *Switch*

Para obter informações mais precisas com relação ao funcionamento do *switch*, entende-se que deve haver, para cada uma das suas interfaces, um nível maior de detalhamento. Por este motivo, foram então desenvolvidas as páginas sobre as conexões do *switch*.

Foram desenvolvidas as páginas de informações sobre as conexões do *switch* somente para algumas interfaces. Elas permitem análise de três conexões, a saber: com a redeUFSC; com o CTC e com o CFM.

Através da página de informações sobre as conexões do *switch* é possível ver as seguintes informações da interconexão: o estado operacional; a velocidade de tráfego; a quantidade de octetos que chegam; a quantidade de octetos que saem; a quantidade de octetos que chegam com erro; e a quantidade de octetos que saem com erro.

No Anexo M, encontra-se o código fonte, em HTML, da página com informações sobre a conexão do *switch*, e no Anexo M1, está o código fonte, em Java, do applet utilizado na referida página. Cabe ressaltar que o mesmo tipo de página foi também desenvolvido para as conexões com o CTC e o CFM.Ver Figura 39.

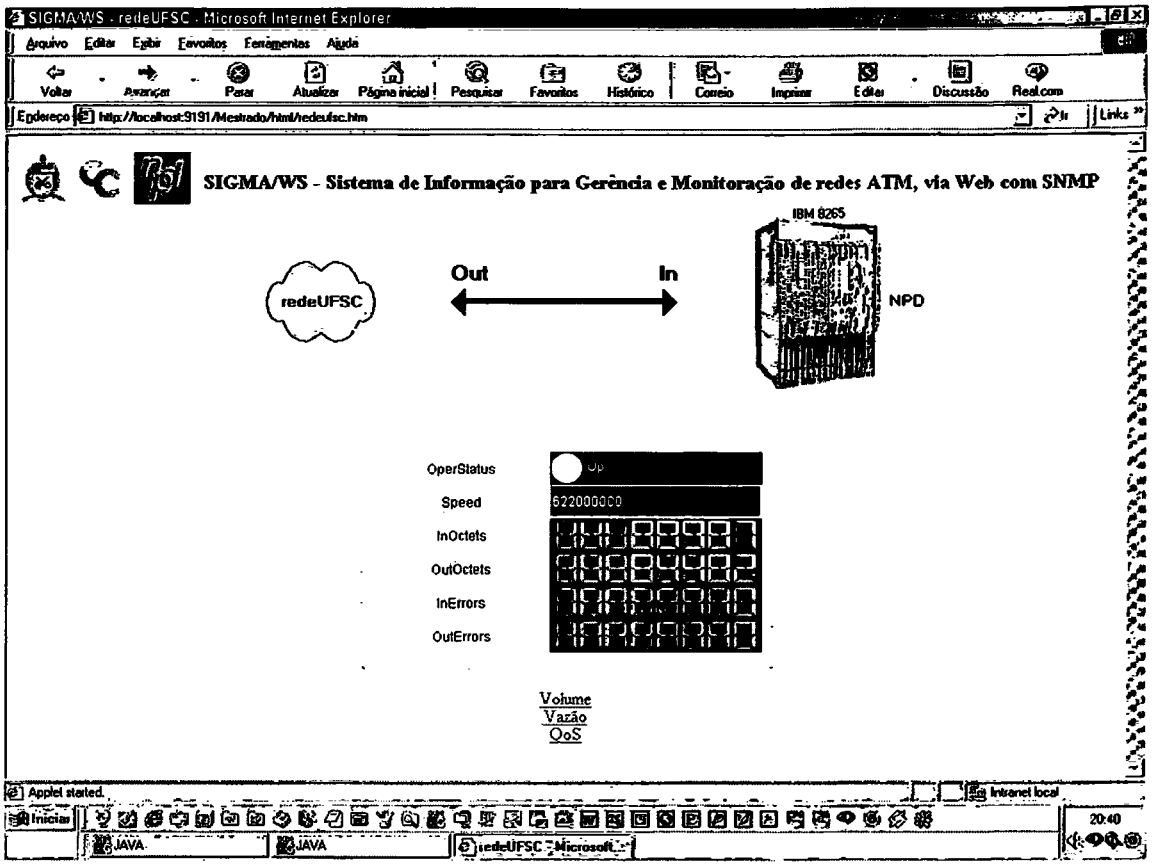

Figura 39: Página de informações sobre a conexão do *switch*.

Conforme mostrado na Figura 39, considera-se como ***in*** os dados que chegam ao *switch* IBM 8265 e como ***out*** os dados que saem dele, uma vez que a informação do tráfego na conexão é obtida através desse *switch*. A seguir, descrevem-se as variáveis utilizadas nesta página.

ifOperStatus é a variável da MIB-II (RFC-1213) que armazena um valor que identifica o estado operacional da interface. Este valor pode ser: 1(Up, representado pela cor verde); 2 (Down, representado pela cor vermelha); e 3 (Testing, representado pela cor azul).

ifSpeed é a variável da MIB-II (RFC-1213) que armazena um valor representativo da largura de banda (em bits por segundos) da interface, ou seja da sua velocidade. Para as interfaces nas quais a largura de banda não varia, ou para aquelas em que esta informação não está disponível, é apresentada a largura de banda nominal do enlace.

ifInOctets é a variável da MIB-II (RFC-1213) que armazena um valor que representa o número de octetos recebidos na interface. O número de octetos transmitidos é obtido através da variável ***ifOutOctets*** da MIB-II (RFC-1213).

ifInErrors é a variável da MIB-II (RFC-1213) que armazena um valor que representa a quantidade de octetos que chegam com erro. Enquanto que, a variável que armazena o número de octetos que saem com erro é ***ifOutErrors***. Cabe observar que a detecção de erros em pacotes previne que eles venham a ser recebidos ou enviados com erro.

Todas estas informações aqui descritas (estado operacional, largura de banda, número de octetos que chegam, número de octetos que saem, número de octetos que chegam com erro e número de octetos que saem com erro) foram reunidas na página de informações sobre a conexão do *switch*, com o objetivo de dar ao usuário da ferramenta uma forma de, sucintamente, ou seja, com poucos dados, ter uma noção do estado da conexão.

Considerando a grande quantidade de ocorrências possíveis na conexão, percebe-se a necessidade de ter maiores informações que permitam uma melhor análise. Com o objetivo de permitir que o usuário acesse com maiores detalhes as informações sobre: volume; vazão; e QoS na interface, foram então desenvolvidas três páginas. A seguir, são descritas estas páginas que permitem monitorar: o volume; a vazão; e a QoS da conexão.

5.2.1 Páginas com informações sobre o Volume na Conexão

O volume em uma conexão ou enlace representa a quantidade de informação transmitida num determinado espaço de tempo. Considerando que as informações chegam e saem do *switch* separadamente, através de cada uma das interfaces, deve-se analisar o tráfego de cada uma das interfaces individualmente. Além disso, para

monitorar o volume de dados transmitidos em uma interface, faz-se necessário monitorar tanto a informação que chega quanto informação que sai.

Quando o tráfego de cada uma das interfaces é monitorado separadamente, torna-se possível determinar quais são as conexões que estão atuando com um volume de tráfego acima ou abaixo da média.

Para cada uma destas interfaces monitoradas de um *switch*, faz-se necessário considerar, separadamente, o volume de tráfego que entra e o volume de tráfego que sai. Tal que seja possível determinar, com segurança, a origem dos gargalos na rede.

Assim sendo, o desenvolvimento da página que apresenta o volume de tráfego numa conexão, está baseado em duas informações: o volume de tráfego que chega até a interface, e o volume de tráfego que sai da interface. Por esse motivo foram implementados dois gráficos, um para cada tipo de informação, permitindo analisar individualmente o tráfego de entrada e o tráfego de saída.

No Anexo N, encontra-se o código fonte, em HTML, da página com informações sobre o volume de tráfego na conexão, e no Anexo N1, está o código fonte em Java do applet utilizado na referida página. O mesmo tipo de página foi também desenvolvido para as conexões com o CTC e o CFM.Ver Figura 40.

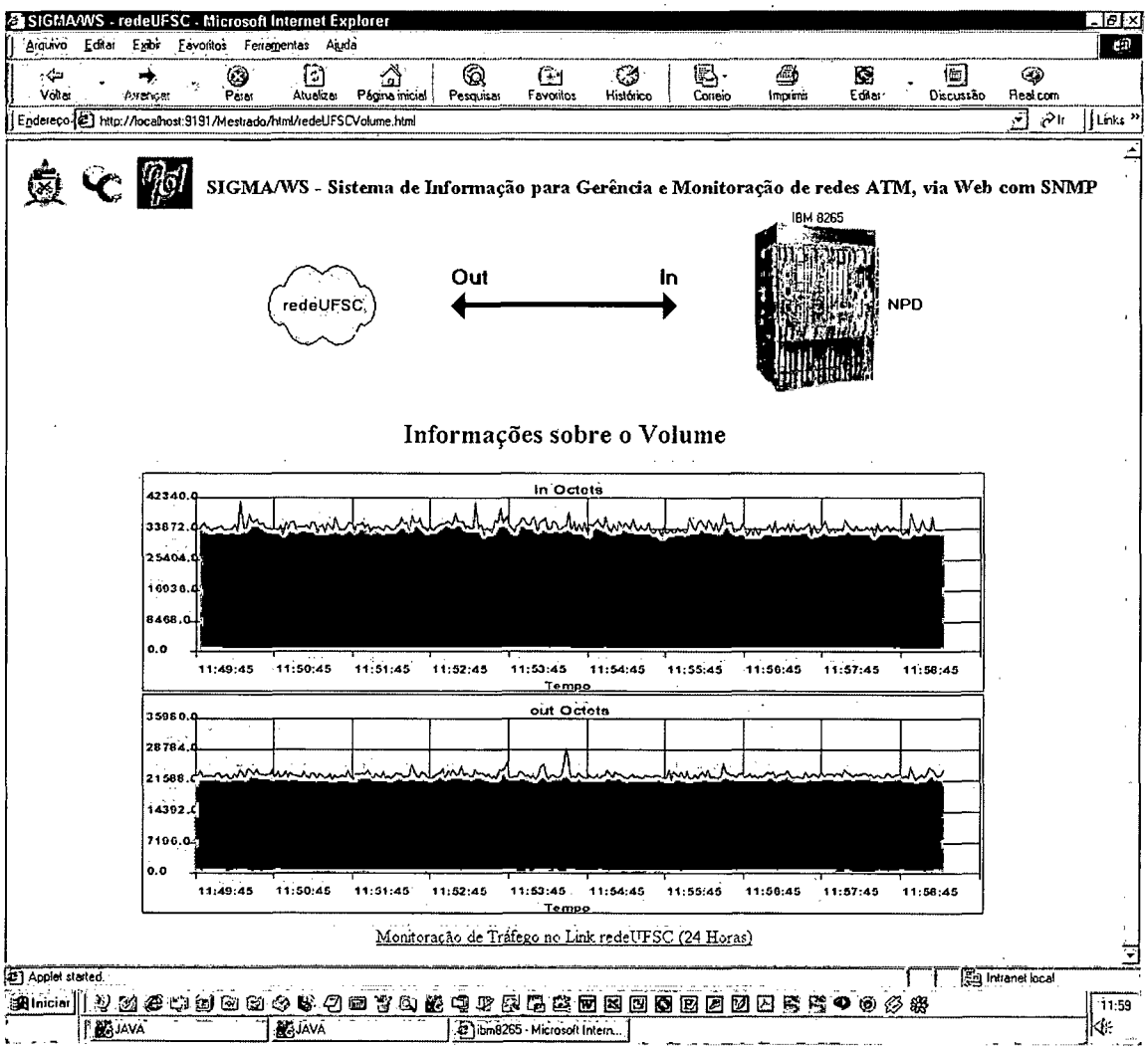

Figura 40: Página de informações sobre o volume na conexão.

Conforme demonstra a Figura 40, a página de informações sobre o volume de tráfego na conexão apresenta dois gráficos. Os gráficos representam o volume de dados que entrou e o volume de dados que saiu da interface.

A informação-base, para a geração do gráfico que apresenta o volume de dados que entrou na interface, é obtida através da variável ***ifInOctets*** da MIB-II (RFC-1213). Este valor expressa o número total de octetos recebidos na interface.

O outro gráfico – que apresenta o volume de dados que entrou na interface, baseia-se no valor da variável ***ifOutOctets*** da MIB-II (RFC-1213), que expressa o número total de octetos recebidos na interface.

Os gráficos apresentados na página de volume da conexão estão desenvolvidos para apresentar o tráfego em tempo real. Uma vez que o usuário entra na página, os dados obtidos do *switch* vão sendo utilizados para a geração do gráfico, desse modo, apresentam ao usuário a realidade do tráfego naquele instante.

Com o objetivo de visualizar em tempo real o volume de octetos transmitidos na conexão, permitiu-se a visualização de um total de 10 (dez) minutos de tráfego. Esse tempo pode ser considerado suficiente para uma verificação rápida do volume de octetos no enlace.

Caso o usuário necessite avaliar o tráfego por um período maior de tempo, ele deve utilizar-se da hiperligação apresentada abaixo dos gráficos. Esta hiperligação executa a abertura de uma nova janela de navegação *Web*, semelhante à página de volume de tráfego na conexão, que gera gráficos que apresentam os dados de 24 horas de monitoração.

No Anexo O, encontra-se o código fonte em HTML, da página com informações sobre o volume de tráfego na conexão (24 h), e no Anexo O1, está o código fonte, em Java, do applet utilizado na referida página. O mesmo tipo de página foi também desenvolvido para as conexões com o CTC e o CFM. Ver Figura 41.

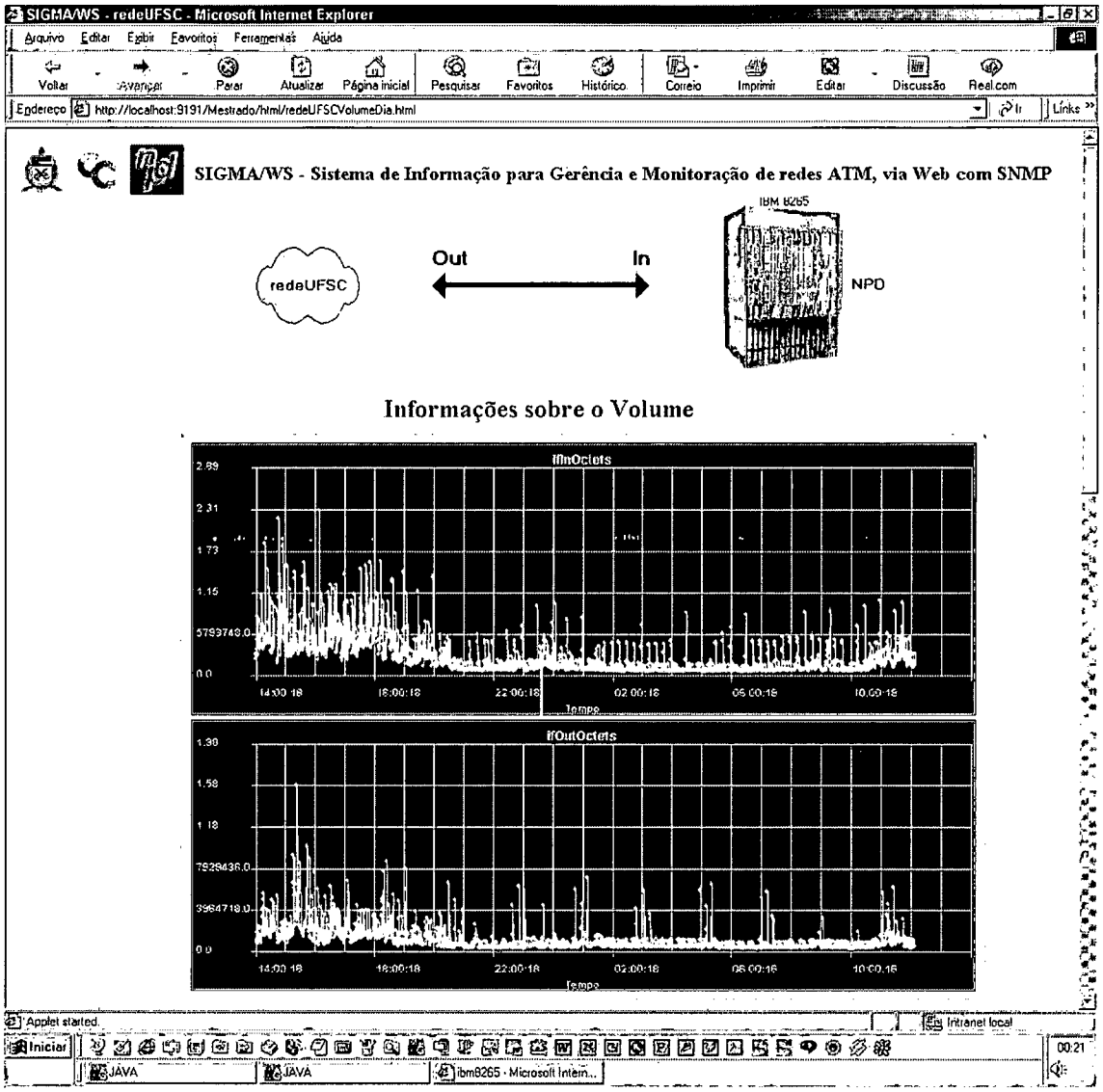

Figura 41: Página de informações sobre o volume na conexão (24 h).

O tipo de informação utilizada para gerar o gráfico de volume de tráfego 24 Horas é o igual ao tipo de informação utilizado para gerar o gráfico de volume de tráfego de 10 minutos, a saber: total de octetos recebidos; e total de octetos enviados em uma determinada interface.

O que os distingue, todavia, é a visualização mais detalhada oferecida no gráfico de 10 minutos, contra uma monitoração mais ampla oferecida pelo gráfico de 24 horas. Pois, para expressar em um gráfico o comportamento do volume de tráfego de um dia inteiro, necessita-se uma amostragem de dados mais esparsa, ou seja, com um intervalo de tempo bem maior.

A página de informações sobre o volume de tráfego na conexão (24 Horas), deve ser utilizada quando se deseja fazer uma análise, da variação do volume de tráfego em um dia, não excluindo a necessidade de um acompanhamento por vários dias seguidos.

5.2.2 As páginas com informações sobre a Vazão na Conexão

A vazão de uma conexão representa o percentual de uso da mesma, ou seja, a relação entre o volume de dados transmitidos em uma conexão e a largura de banda deste mesmo. Determinando desta maneira o percentual de ocupação de uma conexão. Assim sendo, optou por um gráfico com indicações percentuais.

O tipo de gráfico utilizado para demonstrar a vazão foi o de *Gauge*, que demonstra, tal como um medidor o percentual utilizado na conexão. Com o objetivo de chamar a atenção do usuário para diferentes patamares de utilização da banda, foram definidos dois *threshold's*: aos 50% e aos 75%.

Assim sendo, enquanto a vazão for menor que 50% ela é apresentada na cor verde; quando a vazão estiver entre 50% e 74% ela é apresentada na cor amarela e quando a vazão for de 75% ou mais, ela apresenta-se na cor vermelha.

A exemplo dos gráficos que apresentam o volume de tráfego em uma conexão, os gráficos utilizados para expressar a vazão, devem fazê-lo considerando tanto a entrada quanto à saída de dados, separadamente, pelas mesmas razões. Por isso, a página de informações sobre a vazão na conexão foi implementada para apresentar dois gráficos.

No Anexo P encontra-se o código fonte em HTML da página com informações sobre a vazão na conexão, e no Anexo P1 está o código fonte em Java do applet utilizado na referida página. O mesmo tipo de página foi também desenvolvido para as conexões com o CTC e o CFM. Ver: Figura 42.

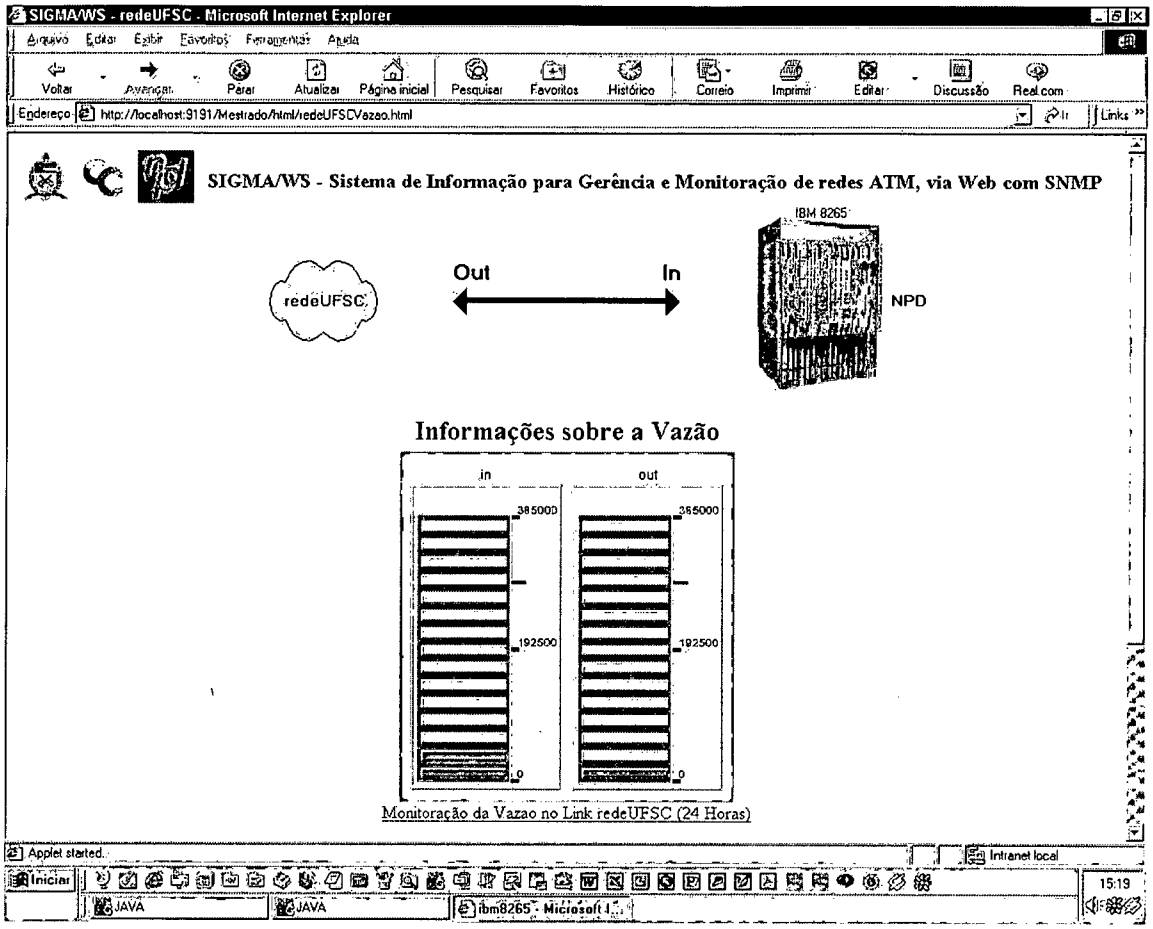

Figura 42: A página de informações sobre a vazão na conexão.

ifInOctets é a variável da MIB-II (RFC-1213) que armazena um valor, o qual expressa o número total de octetos recebidos na interface. Este total de octetos recebidos, que representa o volume, é então relacionado com a capacidade total de transmissão na conexão para que se obtenha a vazão.

O mesmo procedimento ocorre com a vazão de dados de saída, que é obtida através da variável ***ifOutOctets*** da MIB-II (RFC-1213). Essa variável expressa o número total de octetos enviados na interface.

Considerando que a informação apresentada nesta página é gerada em tempo real, atualizando-se automaticamente, sobrepondo sempre a última informação, julgou-se necessário ter uma página que apresentasse um histórico de dados sobre a vazão, para permitir uma monitoração e análise mais abrangente, por meio de uma avaliação sobre um período de tempo maior.

Em função disso, abaixo dos gráficos que apresentam a vazão na interface, é disponibilizada uma hiper-ligação que executa a abertura de uma nova janela para a monitoração da vazão, na interface com a possibilidade de visualizar de 24 horas.

No Anexo Q encontra-se o código fonte em HTML da página com informações sobre a vazão na conexão, e no Anexo Q1 está o código fonte em Java do applet utilizado na referida página. O mesmo tipo de página foi também desenvolvido para as conexões com o CTC e o CFM. Ver: Figura 43.

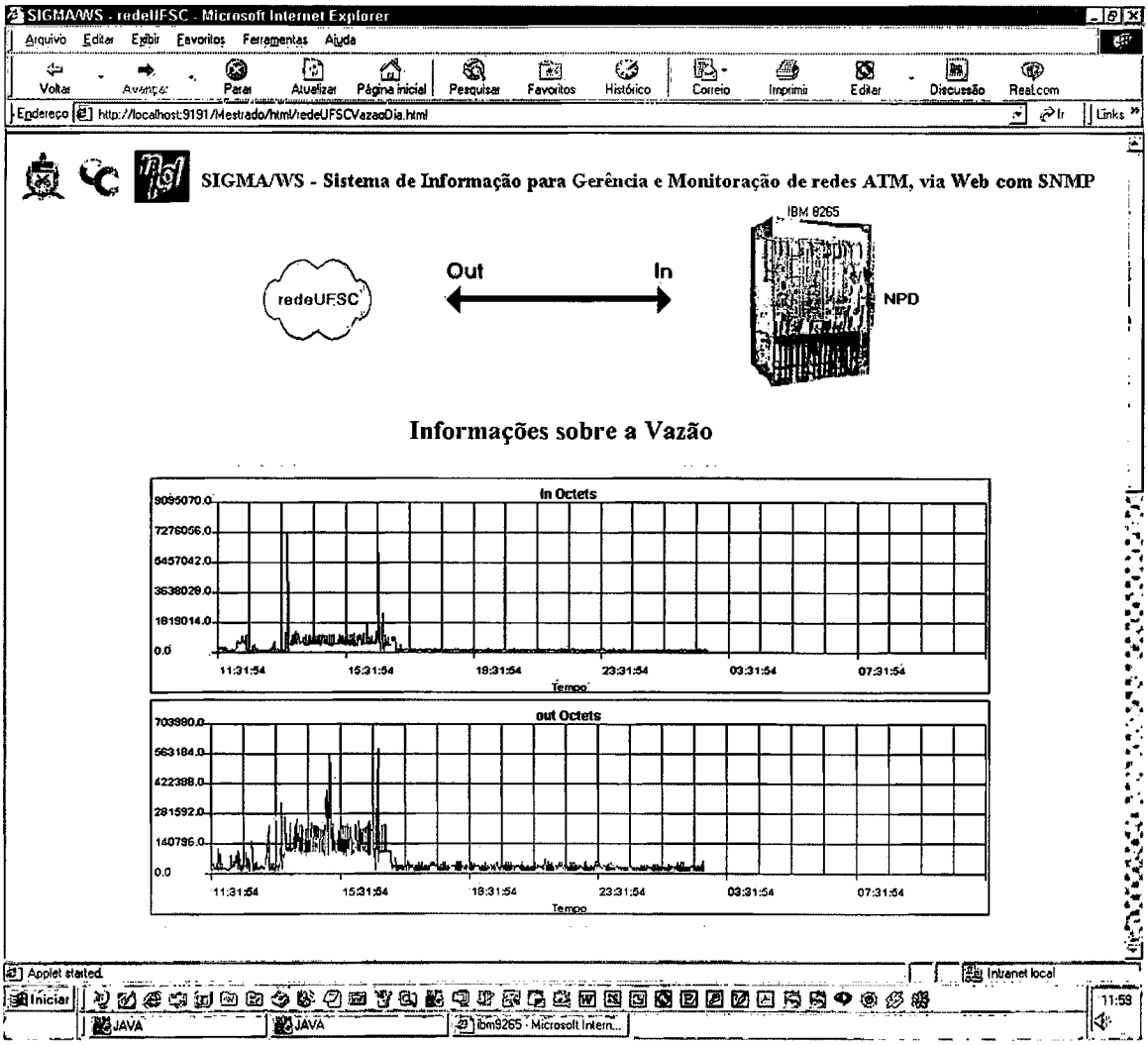

Figura 43: A página de informações sobre a vazão na conexão (24 h).

Com base na monitoração de um dia inteiro torna-se possível identificar horários e/ou situações de congestionamento na rede. Porém, para uma perfeita compreensão da situação da rede, faz-se necessário uma adequada monitoração da rede.

5.2.3 As páginas com informações sobre a QoS na Conexão

Quanto mais heterogêneo é o tráfego em uma rede, mais serviços deverão ser oferecidos para que os usuários tenham uma infra-estrutura de rede compatível com as aplicações que serão executadas na rede. O aumento dos serviços de rede amplia também a necessidade de controles e de monitorações.

A qualidade de serviço (QoS), pode ser vista sob vários aspectos. Se diferentes tipos de tráfego requerem características diferentes de infra-estrutura da rede, então, pode-se assumir que são necessários controles distintos dos serviços de rede, devido à especificidade do tráfego a ser suportado.

As redes ATM destacam-se por sua capacidade de transmitir dados em alta velocidade, aproveitando muito bem os meios de comunicação, tanto com tráfego em rajada, quanto com tráfego contínuo, Graças à sua capacidade de transmissão e a sua orientação para a comutação de pacotes.

Considerando esta forte característica das redes ATM, voltada à velocidade de transmissão, optou-se por desenvolver as páginas de monitoração da qualidade de serviço, controlando a quantidade de erros na transmissão da conexão.

A monitoração da qualidade de serviço, medida individualmente, em cada conexão do *switch* justifica-se, porque o índice de erros pode variar de uma conexão para outra e com a análise individual das conexões isto pode vir a ser constatado.

Assim, a qualidade de serviço vem contribuir para com o objetivo principal da ferramenta, que é oferecer subsídios para melhor gerenciar, monitorar e operar a rede, procurando aumentar a disponibilidade desta, facilitando a detecção de falhas.

As informações disponibilizadas na página de informações sobre as conexões, foram escolhidas tendo-se como objetivo a comprovação da qualidade de serviço nas transmissões da conexão.

No Anexo R encontra-se o código fonte em HTML da página com informações sobre a QoS na conexão, e no Anexo Q1 está o código fonte em Java do applet utilizado na referida página. O mesmo tipo de página foi também desenvolvido para as conexões com o CTC e o CFM. Ver: Figura 44.

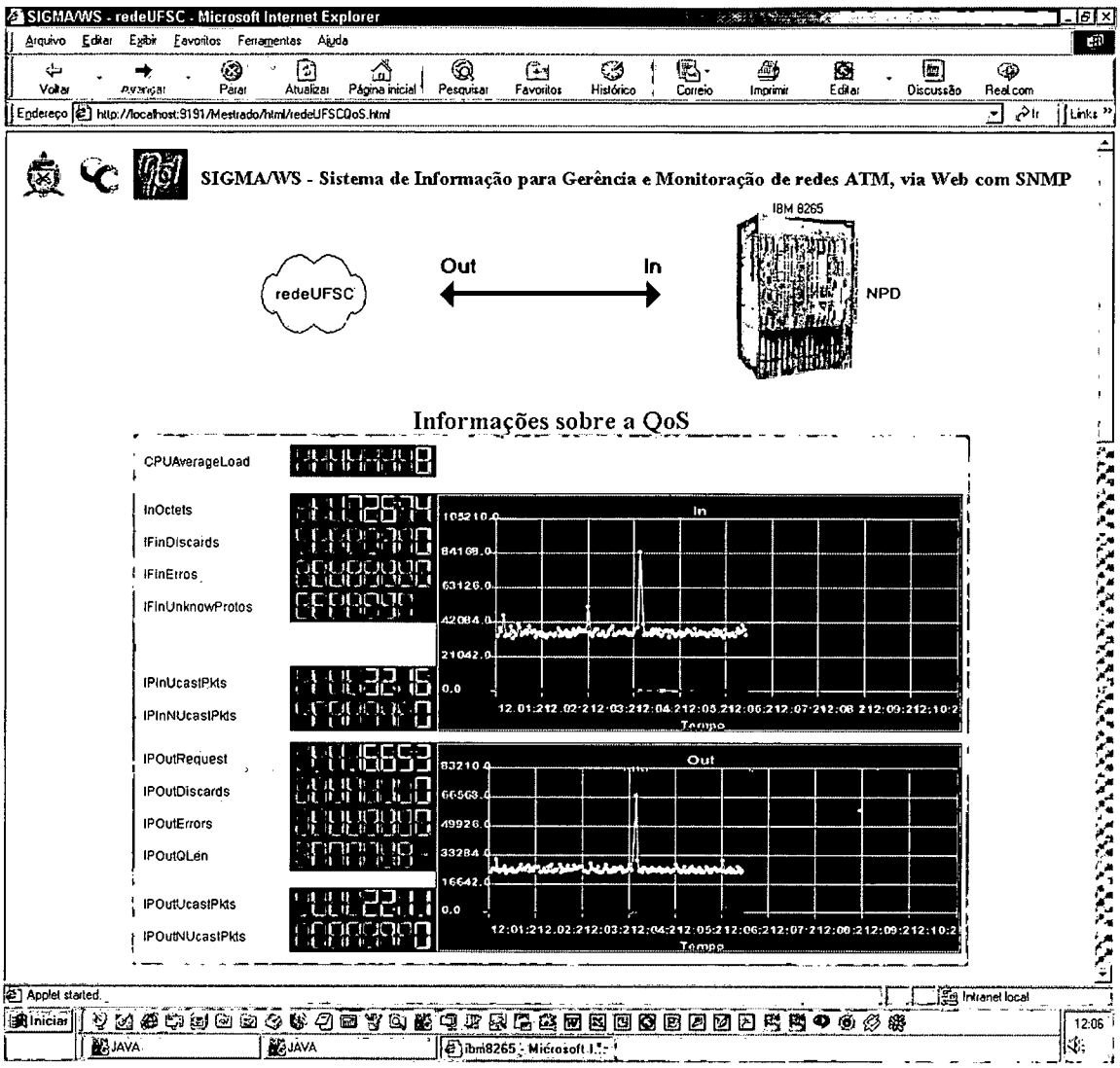

Figura 44: A página de informações sobre a QoS na conexão.

A seguir são identificadas as variáveis utilizadas para a obtenção das informações apresentadas na página de informações sobre a qualidade de serviço na conexão. A exemplo das páginas de Volume e de Vazão, a Qualidade de Serviço (QoS) deve ser analisada considerando separadamente a entrada e a saída de dados, pelas mesmas razões.

Os fatores considerados mais importantes para a avaliação da qualidade de serviço na transmissão da conexão foram: o tempo de ocupação do processador; o volume de pacotes transmitidos; o volume de pacotes descartados, o volume de pacotes com erros, o volume de pacotes com erro por protocolo não conhecido.

cpuAverageLoad é a variável da MIB proprietária do IBM 8265 versão 4.12, que armazena a informação do tempo de carga, ou seja, de uso da CPU. O valor desta variável pode estar entre: 0 (zero) e 100 (cem). Um valor 0 (zero) indica que o processador do *switch* esteve sempre disponível. Um valor 100 (cem) indica que o processador nunca esteve disponível (ou seja, esteve sempre ocupado).

Como o *switch* possui um único processador, existe somente um valor representando sua utilização. Este valor representa o tempo de uso do processador pelo *switch*, para processar o tráfego de todas as interfaces.

ifInOctets é a variável da MIB-II (RFC-1213) que armazena um valor que representa o volume de pacotes recebidos. Esta variável é utilizada em outras páginas, mas, no contexto da qualidade de serviço, é útil como base para a comparação entre o tráfego recebido com êxito e o tráfego não retransmitido ou recebido com erro.

ifInDiscards é a variável da MIB-II (RFC-1213) que armazena um valor que representa o volume de pacotes recebidos pela interface e descartados.

ifInErrors é a variável da MIB-II (RFC-1213) que armazena um valor que representa o volume total de pacotes recebidos que possuíam erros.

ifInUnknowProtos é a variável da MIB-II (RFC-1213) que armazena um valor que representa volume de pacotes que deram erro por usar um protocolo desconhecido.

A soma das variáveis: ***ifInDiscards***; ***ifErrors***; ***ifInUnknowProtos*** representam volume total de pacotes recebidos e não encaminhados.

IfInUcastPkts é a variável da MIB-II (RFC-1213) que armazena um valor que representa o número de pacotes *unicast* enviados a um protocolo de uma camada de nível superior.

IfInNUcastPkts é a variável da MIB-II (RFC-1213) que armazena um valor, o qual representa o número de pacotes *non-unicast* (i.e., *sub network broadcast* ou *sub network multicast*) entregue a um protocolo de nível superior.

ifOutOctets é a variável da MIB-II (RFC-1213) que armazena um valor que representa o volume de octetos transmitidos. Esta variável, – a exemplo da variável ***ifInOctets***, também utilizada em outras páginas, serve nesta página como base para a comparação entre o tráfego recebido com êxito e o tráfego não retransmitido ou recebido com erro.

ifOutDiscards é a variável da MIB-II (RFC-1213) que armazena um valor que representa o volume de octetos a serem transmitidos e que foram descartados.

ifOutErrors é a variável da MIB-II (RFC-1213) que armazena um valor que representa o volume total de octetos a serem transmitidos e que não foram em virtude de erros.

IfOutQLen é a variável da MIB-II (RFC-1213) que armazena um valor que representa o comprimento da fila de pacotes de saída (em pacotes).

IfInUcastPkts é a variável da MIB-II (RFC-1213) que armazena um valor que representa o número total de pacotes *unicast* que os protocolos de camadas superiores requisitaram, incluindo aqueles que foram descartados ou não transmitidos.

IfInNUcastPkts é a variável da MIB-II (RFC-1213) que armazena um valor que representa o número total de pacotes *non-unicast* (i.e., *sub network broadcast* ou *sub network multicast*) que os protocolos de camadas superiores requisitaram, incluindo aqueles que foram descartados ou não transmitidos.

A opção por estas variáveis é feita com o intuito de demonstrar a qualidade das transmissões na conexão, utilizando a relação entre as transmissões efetuadas com

sucesso e as transmissões com problemas: de transmissão; de protocolo; ou de empacotamento.

Para auxiliar a avaliação desta relação é que os gráficos foram apresentados. O tipo de gráfico utilizado para apresentar a QoS é o de *multi-linhas*, por permitir a demonstração, de diversas variáveis ao mesmo tempo. Observe-se que as cores utilizadas para indicar os valores das variáveis servem como legenda para as cores no gráfico.

A avaliação da qualidade de serviço é feita considerando separadamente a entrada e a saída, porque problemas com protocolos não conhecidos podem ser gerados por uma única máquina, atingindo um único sentido na conexão.

Capítulo 06: Experimentos e Testes Realizados

Capítulo 06: Experimentos e Testes Realizados

O ambiente *Web* é do tipo *Cliente-Servidor*. Por um lado temos os navegadores *Web* e por outro os servidores *Web*. Os navegadores fazem as solicitações, os servidores as processam e as respondem.

Assim sendo, para que um usuário de qualquer ponto da rede, utilizando um navegador *Web*, possa acessar a ferramenta SIGMA/WS e as suas páginas, estas devem estar armazenadas em algum servidor *Web*, capaz de responder às solicitações dos navegadores *Web*.

Por motivos de segurança, o computador que hospeda as páginas HTML e os applet's Java da ferramenta SIGMA/WS – atuando como um servidor *Web*, deve ter um endereço IP estático para poder ser configurado na *community* dos *switch's* e ter acesso aos mesmos.

O *Management Builder* da Adventnet – programa utilizado no desenvolvimento da ferramenta SIGMA/WS, possui dois aplicativos que foram necessários e úteis nos testes e experimentos. Um servidor *Web* (*StartWebServer*) e um servidor de *Applet*'s SNMP (*StartSasServer*).

6.1 Procedimentos seguidos para efetuar os experimentos e testes

O mesmo computador utilizado para o desenvolvimento da ferramenta, e no qual as páginas HTML e os *applet's Java* já estavam armazenados, foi também utilizado como servidor.

Uma vez que este computador tinha como sistema operacional o Windows98, foi então necessário utilizar o Servidor *Web* da Adventnet (*StartWebServer*), durante os experimentos e testes. Assim, antes de iniciar o teste é necessário executar o *StartWebServer*. Ver Figura 45.

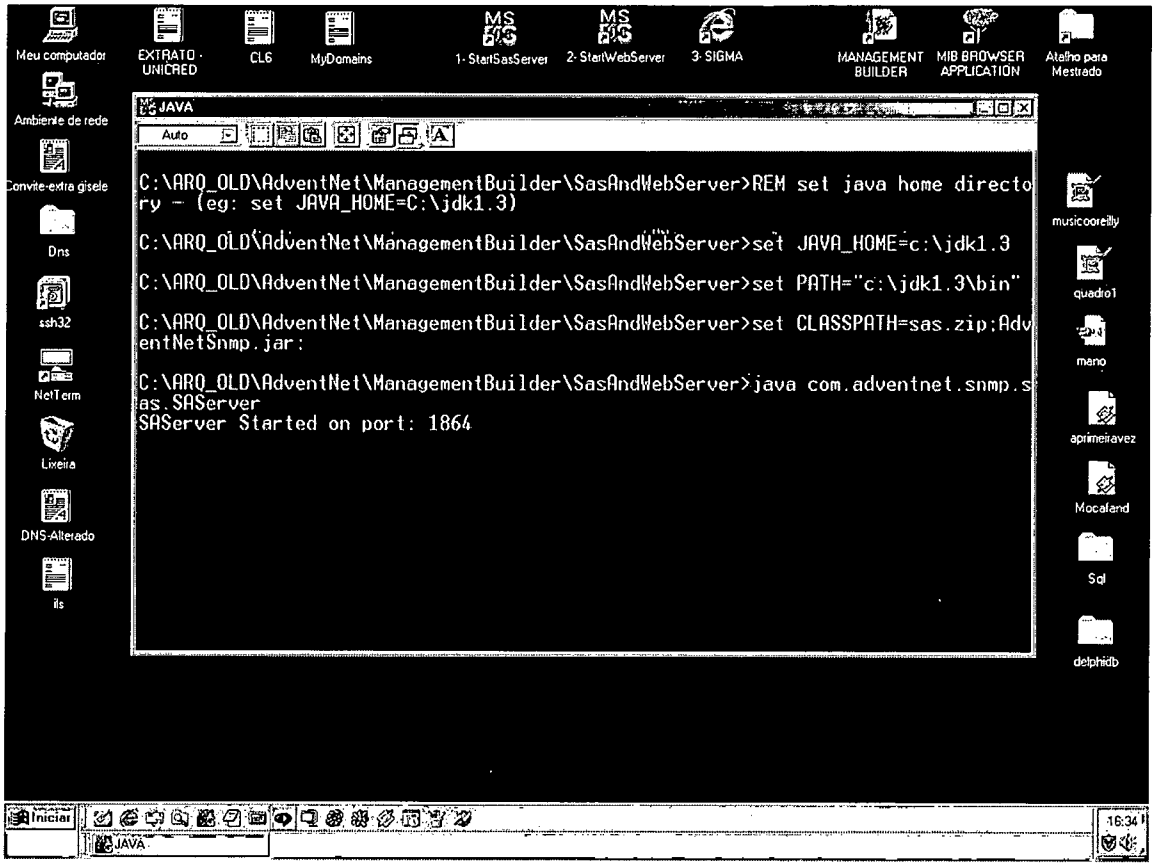

Figura 45: O utilitário *StartWebServer* em execução.

Além de ter um Servidor *Web* sendo executado – para responder as solicitações de navegadores *Web*, faz-se necessário também que o Servidor de *Applet's* SNMP (*StartSasServer*) também esteja sendo executado. O Servidor de *Applet's* SNMP, é responsável pela correta execução dos *Applet's Java* que estão inseridos nas páginas da ferramenta SIGMA/WS. Ver Figura 46.

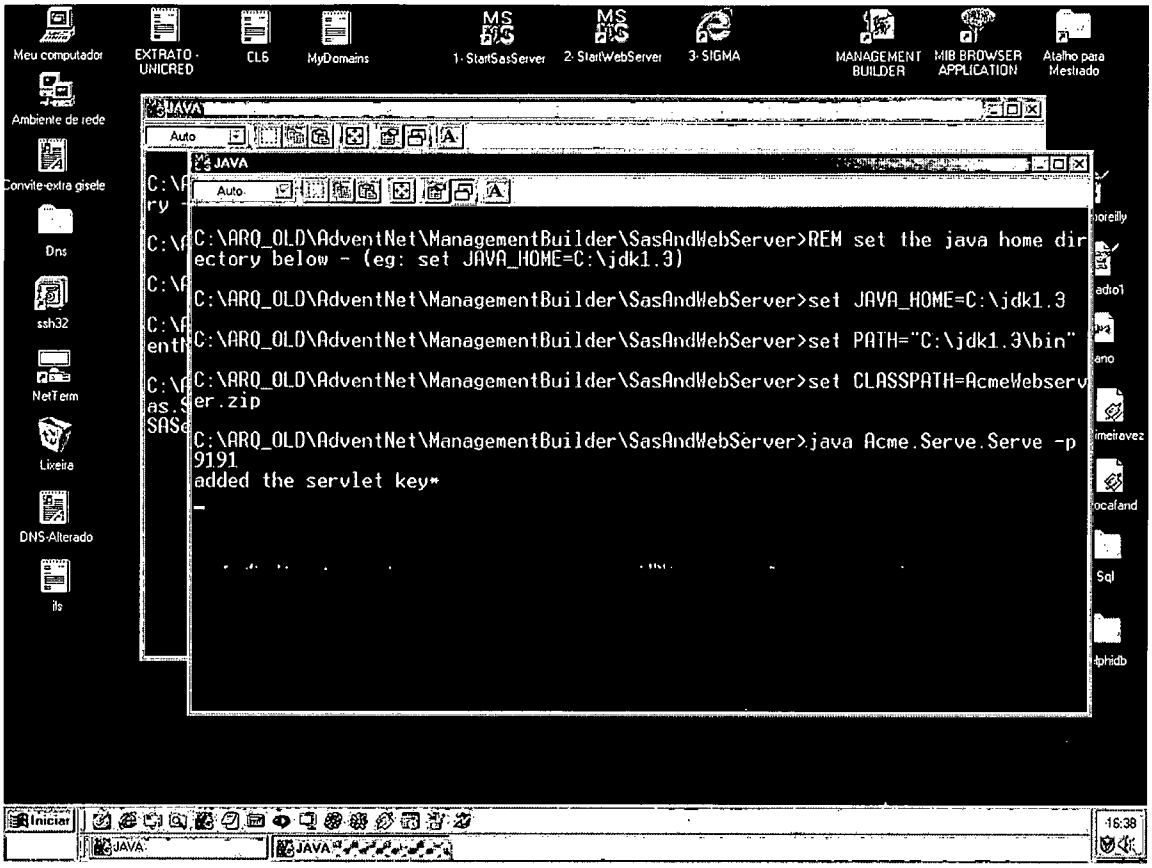

Figura 46: O utilitário *StartSasServer* em execução.

A partir do momento que estes dois utilitários (*StartWebServer* e *StartSasServer*) estão sendo executados, o computador está pronto para funcionar como Servidor *Web*. Utilizando-se um navegador *Web* é possível acessar as páginas da ferramenta SIGMA/WS. Para testar a ferramenta SIGMA/WS, faz-se necessário executar um navegador *Web* e acessar a página da ferramenta (através do endereço desta).

Durante o período em que se estava implementando a ferramenta, os testes foram feitos de dentro da redeUFSC, mais especificamente da sala de estudos do CPGCC onde a ferramenta foi desenvolvida.

Por estar o navegador (*Cliente Web*) sendo executado no mesmo *host,* ou seja, no mesmo computador que o servidor (*Servidor Web*) pode-se utilizar o endereço *localhost*, para acessar a ferramenta. Ver Figura 47.

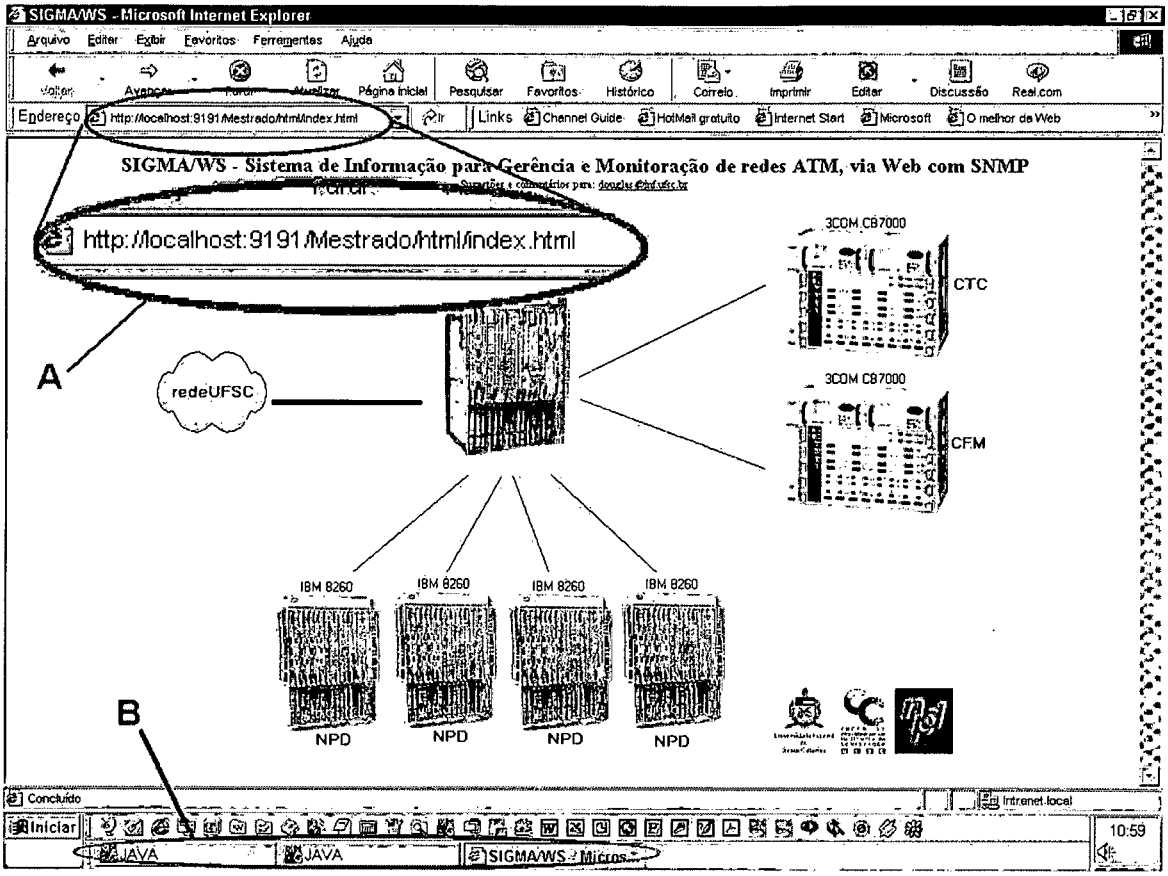

Figura 47: Acessando a SIGMA/WS do computador-servidor.

Conforme apresenta a Figura 47 no detalhe "A", o endereço utilizado para acessar a página é o *localhost*. Pode também observar no detalhe "B" que os programas: Servidor *Web* (*StartWebServer*) e Servidor de Applet's SNMP (*StartSasServer*), estão sendo executados juntamente com o navegador *Web*.

Por questões de segurança, este método somente é recomendado para a fase de implementação da ferramenta, pois o computador que estiver atuando como servidor da ferramenta SIGMA/WS, tem direito de acesso a *community* dos *switch's* gerenciados, e deve ter o acesso ao seu uso em console restrito.

Na concepção da ferramenta SIGMA/WS, julgou-se adequado para o seu funcionamento que um computador atuasse como o Servidor *Web* da ferramenta, para outros computadores pudessem acessá-lo pela rede Internet.

Caso o Servidor *Web* seja um *host* com nome de domínio registrado, poderia ser acessado por seu nome. Todavia, os acessos às páginas da ferramenta, sempre estão disponíveis se o usuário fizer uso do endereço IP do *host.* Ver Figura 48.

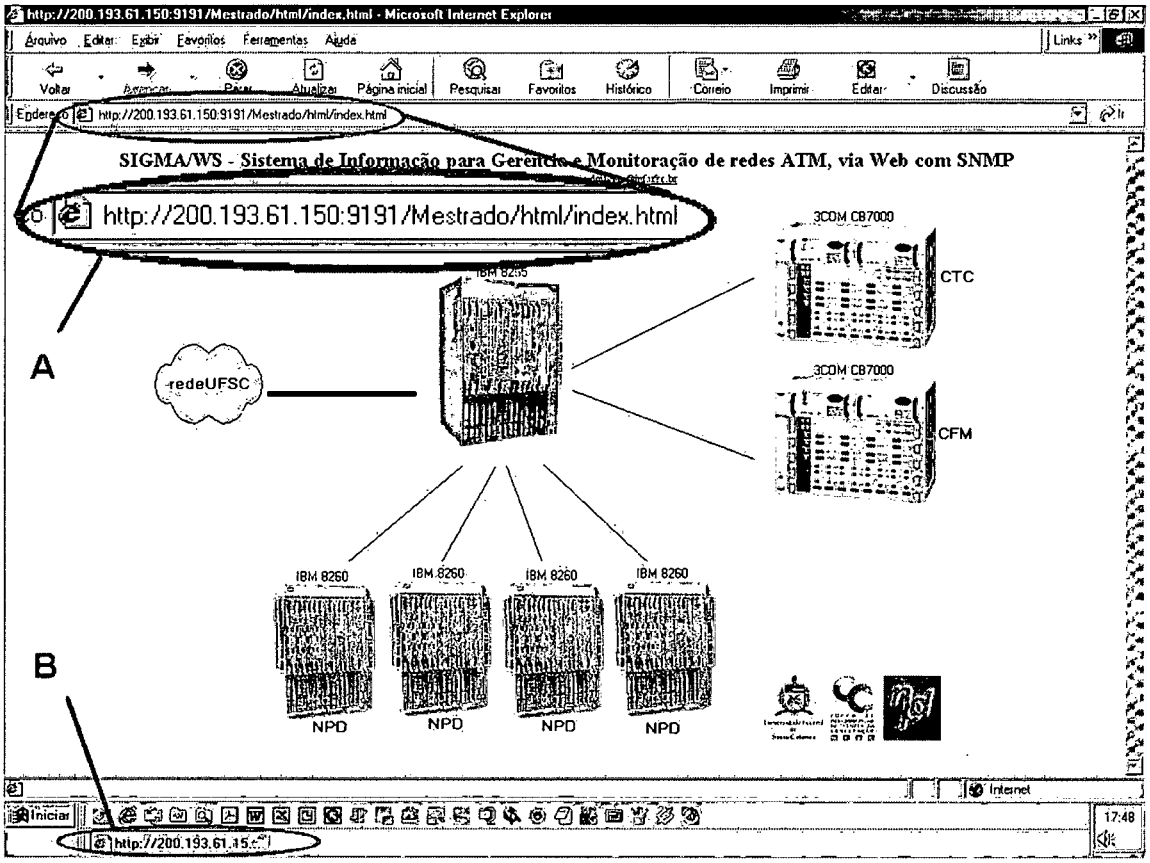

Figura 48: Acessando a SIGMA/WS de um computador-cliente.

Conforme apresenta a Figura 48 no detalhe "A", o endereço utilizado para acessar a página é o endereço IP do computador que hospeda a página (200.193.61.150). Pode-se também observar no detalhe "B" que somente o navegador *Web* está sendo utilizado neste computador (*Cliente Web*).

Depois que a fase de desenvolvimento terminou passou-se a testar a ferramenta SIGMA/WS e o seu uso, remotamente, da cidade de Lages. Durante os testes e experimentos, pôde-se constatar também que, além de ser possível acessar a ferramenta SIGMA/WS por um cliente *Web* remotamente situado (fora da rede local, mas dentro da Internet). É também possível que o Servidor *Web* da ferramenta esteja fora da rede local na qual estão os equipamentos que serão monitorados.

Durante a implementação foram escolhidas as variáveis e foram efetuados testes de monitoração. Depois de terminada a implementação, iniciaram-se os experimentos. O objetivo nesta etapa do desenvolvimento, era verificar a utilidade e a usabilidade da ferramenta, verificando qualidades e defeitos.

Aprendizado contínuo

Logo que as monitorações se iniciaram, percebeu-se que o conhecimento da rede, seu tráfego, e suas aplicações é imprescindível para o usuário da ferramenta. Uma vez que, sem isso faltaria base para determinar se os valores medidos (volume; vazão; octetos recebidos; octetos enviados; etc.) eram "normais" ou não.

No início, somente as páginas que apresentam gráficos permitiam uma avaliação sobre gargalos ou subutilização, apresentando a vazão nas conexões. Com o tempo e com conhecimento do tráfego, pôde-se concluir que mesmo as páginas que apresentam dados resumidos – que podem até vir ser consideradas por demais sucintas para alguém que desconheça a rede, podem dar informações suficientes para que alguém que conheça o tráfego da rede possa analisar o estado da mesma.

No dia em que as monitorações foram efetuadas pela primeira vez observou-se que o tráfego, tanto de entrada como de saída no enlace com a redeUFSC era em torno de 400.000 octetos. Porém nos dias que se seguiram o volume de tráfego neste enlace oscilou (horário compreendido entre cinco horas e seis horas da tarde, em torno de 20.000 octetos).

Passou-se então a considerar a hipótese de um erro na monitoração inicial, porém no mesmo dia da semana seguinte, o tráfego voltou a oscilar em torno de dos 400.000 octetos. Em uma verificação com NPD, foi observado que naquele dia (da semana) e naquela hora (do dia) semanalmente realizava-se uma videoconferência pela rede.

Assim, constatou-se, com o passar dos dias de monitoração e observação, que alguns dados que de início nada representavam, ou que simplesmente não faziam sentido, aos poucos foram sendo compreendidos, com o auxílio da ferramenta.

Compreensão do Fluxo de Dados

A noção do volume de tráfego existente em uma conexão é uma informação importante, pois permite analisar o tráfego do enlace através de contínuas monitorações. Porém para compreender o impacto de uma aplicação no desempenho da rede, é necessário analisar conjuntamente várias conexões.

Uma vez percebido – através da monitoração do volume na conexão, o impacto no tráfego da rede gerado pela aplicação de videoconferência. E, observando-se que uso da aplicação tinha reflexos em diversas conexões. Confirmou-se a utilidade da página de informações sobre o tráfego nas conexões. Ver Figura 49.

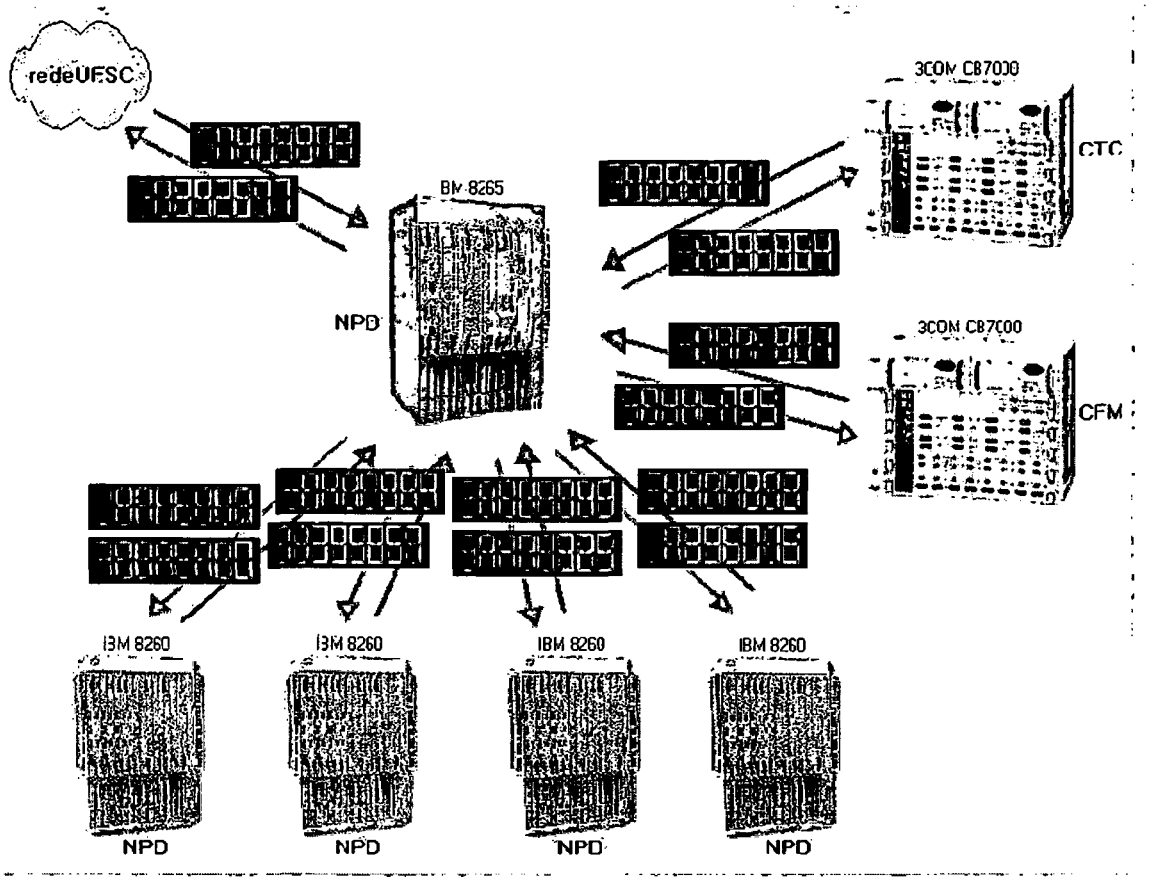

Figura 49: Tráfego nas conexões durante a videoconferência.

A Figura 49, obtida num momento em que na rede ocorre uma das reuniões semanais de teleconferência, percebe-se claramente como o tráfego gerado pela aplicação de teleconferência influencia na rede.

No caso da monitoração apresentada na Figura 49, observa-se que os enlaces pelos quais o tráfego da teleconferência não passa, o volume de octetos não atinge a 1.000 octetos. Por outro lado os enlaces nos quais o tráfego da teleconferência passa, o volume gira em torno de 395.000 octetos.

Outra característica do tráfego da rede, observada durante as monitorações através da ferramenta, foi o sentido do tráfego entre diferentes dias da semana. Em dias úteis, observou-se claramente um volume maior de dados entrando, enquanto que em feriados e finais de semana o volume de dados saindo apresentava-se maior. Ver Figura 50.

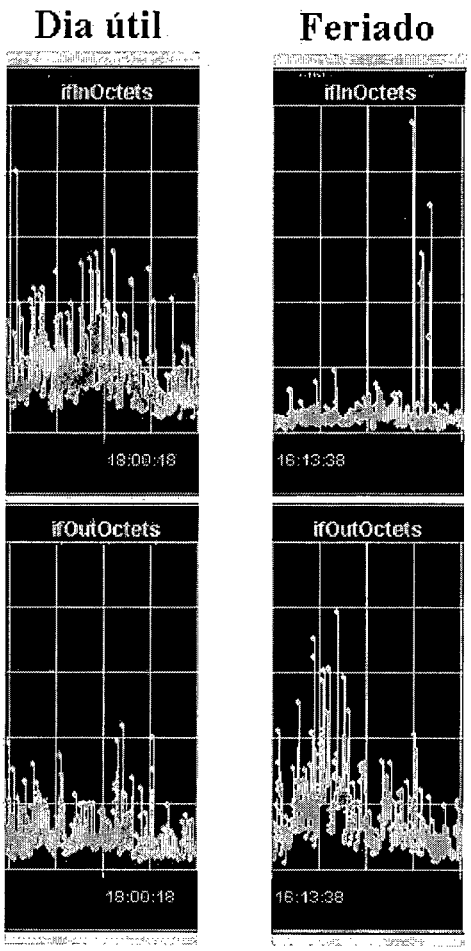

Figura 50: Diferença no sentido do tráfego, em dias úteis e não-úteis.

Caso a ferramenta não apresentasse os dados separando-os de acordo com o sentido do tráfego, esta análise dificilmente poderia ser feita. E esta característica do tráfego não seria observada.

Análise Rápida do Estado da Rede

Todas as vezes – durante os testes e monitorações, em que a ferramenta era utilizada, uma das primeiras páginas a ser visitada era a página de informações sobre o estado das conexões. Essa página é apresentada com a noção gráfica dos enlaces em funcionamento e dos enlaces que não estão funcionando. Assim, através de uma informação visual era possível saber o estado de todas as conexões do *switch*.

Uma Ferramenta para Prevenção

Durante as monitorações, foi verificado também que situações inesperadas podem acontecer uma vez que o *Cliente Web* pode estar acessando de qualquer lugar. Por exemplo: uma monitoração feita da cidade de Lages, a partir de um micro conectado ao *backbone* da TELESC em um determinado dia não pôde ser feita, pois havia problema na conexão redeUFSC-TELESC. Neste caso, se a monitoração fosse feita de dentro da redeUFSC não haveria dificuldade para fazê-la.

Saber se um determinado enlace está ou não em funcionamento ou se o tráfego flui em ambos os sentidos ou ainda se o volume de dados é normal, são informações importantes e que nos dão respostas imediatas, porém, só serão possíveis sempre que o acesso *ClienteWeb-ServidorWeb* estiver disponível e o acesso *ServidorSas-EquipamentoMonitorado* também.

Portanto, para analisar falhas, este tipo de tecnologia nem sempre poderá ser útil. Porém, esta é uma situação na qual somente uma rede paralela e independente (mas muito mais onerosa e ainda assim não imune à falhas) permitiria contornar. Desde que o problema fosse de enlace, e não de equipamento.

Capítulo 07: Conclusão

Capítulo 07: Conclusão

Desenvolveu-se este trabalho tendo como objetivo principal a análise da utilização do ambiente *Web* como uma infra-estrutura para a implementação de uma ferramenta de auxílio na gerência e na monitoração de redes ATM.

Primeiramente as redes de computadores foram estudadas analisando-se os conceitos relevantes para o desenvolvimento do trabalho. Este estudo permitiu compreender melhor as diferenças entre a tecnologia tradicional de redes (*ethernet*) e a tecnologia ATM, comparando-as.

A seguir foi dada ênfase no estudo das redes ATM, suas características inovadoras e os aspectos importantes de sua tecnologia, tendo como objetivo a busca por um nível de conhecimento que permitisse a definição e a implementação adequada de uma ferramenta para a gerência de redes ATM.

Após este estudo sobre as redes de computadores e em especial as redes ATM, começou-se a estudar a gerência das redes de computadores como um instrumento de auxílio na manutenção e no melhor uso das redes.

O estudo de gerência de redes foi relacionando com a tecnologia Internet e mais especificamente com o ambiente *Web*, com a linguagem de programação Java, e também com o protocolo de comunicação SNMP.

A opção pela rede Internet como infra-estrutura básica de estudo e de implementação do trabalho tem fundamento na inegável importância da Internet na atualidade e nas amplas perspectivas de seu crescimento. Como conseqüência, obtém-se aplicabilidade para a ferramenta de gerência de rede que foi desenvolvida.

Na Internet, o uso do ambiente *Web* acaba se tornando a opção mais viável para uma ferramenta de gerência de rede, porque dentre os protocolos da Internet, o

protocolo HTTP é o que melhor se adapta à implementação de uma ferramenta, uma vez que este protocolo têm boas características de programabilidade, interoperabilidade e usabilidade.

A linguagem de programação nativa para o desenvolvimento de páginas hipertexto – que virão a ser utilizadas através do protocolo HTTP – é HTML, que foi utilizada como base para o desenvolvimento das páginas neste trabalho.

Para permitir que a ferramenta pudesse comunicar-se com os elementos gerenciados optou-se por utilizar o protocolo SNMP, por ser mais difundido entre os equipamentos de rede e por não ser proprietário.

Assim sendo, foi necessário buscar uma maneira de embutir códigos de comandos SNMP dentro das páginas, ou seja, dentro do código em HTML. Por isso, a linguagem de programação *Java* foi utilizada, uma vez que ela permite que sejam desenvolvidos *applet's*, que são códigos em *Java* executados através de uma chamada efetuada de dentro do código HTML.

Com o objetivo de desenvolver os *applet's Java* que contivessem o código necessário para executar as chamadas SNMP buscou-se um recurso para a implementação da ferramenta proposta neste trabalho. Esse recurso foi o *Management Builder* da AdventNet.

Uma vez definida a infra-estrutura (rede Internet) e as tecnologias a serem utilizadas (*Web*, HTML, Java e SNMP), foi possível iniciar os trabalhos práticos. Para tanto foi solicitado ao NPD da UFSC que fosse disponibilizado um *switch* ATM com o qual pudessem ser feitas monitorações. Desse modo ficou definido o ambiente de teste para o desenvolvimento da ferramenta.

O *switch* disponibilizado, um IBM 8265, possuía enlaces com a redeUFSC, o CTC, o CFM e alguns enlaces temporários de teste com o NPD. Nesse ambiente foram feitos os testes e as monitorações.

Para que a implementação pudesse ser feita, utilizou-se um computador da sala de estudos do CPGCC da UFSC, onde algumas adaptações tiveram que ser feitas para que o mesmo pudesse ser utilizado.

Inicialmente foi alterada a configuração de rede do computador, para que o mesmo passasse a ter um endereço IP estático. Assim, ele poderia ser cadastrado em uma *community* no *switch* de teste, permitindo que fosse identificado e autorizado pelo *switch*.

A seguir foi instalado no computador o Management Builder da AdventNet para o desenvolvimento da ferramenta. A implementação da ferramenta teve início com a definição de seu lay-out, e com o desenvolvimento de sua página inicial (*home-page*) em HTML. As páginas acessadas a partir dela, e seus respectivos link's, foram então desenvolvidas.

Cada um dos *applet's* utilizados nas diversas páginas da ferramenta foi desenvolvido isoladamente. Os testes iniciais dos *applets* foram efetuados a partir do recurso de desenvolvimento Management Builder, que permite testar os *applet's* em tempo de desenvolvimento.

Nesses testes foram feitas várias monitorações do switch, buscando verificar a eficácia dos *applet's* na busca de informações do *switch*. Ao longo da implementação o tempo de duração dos testes e da monitoração foi sendo aumentado para buscar a viabilidade de uso real dos *applet's*.

Após a implementação de cada *applet* e a verificação da sua eficácia, a chamada ao *applet* recém-desenvolvido era feito na respectiva página *Web*. Então, os testes

passavam a serem feitos fora do Management Builder, ou seja a partir do navegador *Web*.

Para tanto foram utilizados os dois servidores que acompanham o Management Builder, desenvolvidos para testes desse tipo: o Servidor *Web* (*StartWebServer*) e Servidor de *Applet's* SNMP (*StartSasServer*).

Após o término do desenvolvimento de todos os *applet's* que compõem a página, iniciou-se o teste de uso da ferramenta SIGMA/WS. Durante duas semanas a ferramenta foi utilizada para verificar o tráfego do ambiente de teste, monitorando as alterações e comparando dados para observar a real contribuição oferecida pela ferramenta ao gerente da rede, em seu trabalho diário.

7.1 Interpretação dos Resultados

Com base nas monitorações e experimentos realizados, tanto durante a fase de desenvolvimento da ferramenta, com após o término da mesma, foi possível chegar a determinadas conclusões, que são descritas a seguir.

Aprendizado contínuo

A maioria das informações apresentadas pela ferramenta durante uma monitoração, devem ser confrontadas com outras monitorações. Desta maneira, o usuário da ferramenta precisa fazer várias monitorações para que uma análise coerente possa ser feita.

Porém, com o tempo de uso, o usuário responsável pela monitoração, pode vir a adquirir um "conhecimento" sobre a rede e o comportamento de seu tráfego, facilitando a detecção e a prevenção de problemas (tais como gargalos). Por outro lado, este tipo de conhecimento oferece aos responsáveis pela rede, condições de planejar adequadamente a expansão da mesma.

Portanto, com relação ao uso da ferramenta desenvolvida neste trabalho, pode-se considerar que ela contribui para a administração e para a gerência da rede, por ampliar os conhecimentos do administrador da rede.

Além disso, com o conhecimento da rede e de seu tráfego é possível para o administrador da rede determinar, a partir de uma informação simples, tal como o volume de tráfego em uma conexão, se o enlace está tendo um volume "normal" de tráfego ou não, tornando mais rápida a análise do estado da rede.

Análise Rápida do Estado da Rede

Uma rede ATM baseia-se nos enlaces existentes entre os *switch's*. Estando estes disponíveis para a monitoração, torna-se muito fácil utilizar a ferramenta SIGMA/WS para monitorar e analisar as conexões de um *switch* e, conseqüentemente, o estado da rede.

A informação sobre o estado das conexões – da forma como é apresentada pela ferramenta, dispensa o conhecimento sobre a estrutura da rede para que sejam compreendidos os dados apresentados, diminuindo o tempo de diagnóstico.

Compreendendo o Fluxo de Dados

Com as informações apresentadas nas páginas de informações sobre o tráfego nas conexões e informações sobre o volume de tráfego nas conexões, pode-se observar as mudanças no volume e no sentido do tráfego em diferentes dias. Pode-se ainda avaliar o impacto de uma aplicação de alto tráfego sobre a rede, constatando-se a facilidade oferecida pela ferramenta, para a compreensão do tráfego da rede monitorada.

Uma ferramenta para prevenção

Durante as monitorações com o uso da ferramenta e verificando-se a sua usabilidade e a sua utilidade, observou-se que ela serve para a monitoração do funcionamento da rede e para a análise do seu estado (o tráfego, o volume ou QoS), dentro de um escopo de monitoração preventiva, ou seja, antes que um defeito ocorra. Destacando-se a compreensão do tráfego da rede, que facilita muito a administração, o planejamento e a expansão da mesma.

7.2 Alcance dos Objetivos

Pode-se dizer que este trabalho alcançou os objetivos propostos. O ambiente *Web* foi empregado como infra-estrutura para a implementação da ferramenta SIGMA/WS de auxílio na gerência de redes ATM.

Foram utilizadas a linguagem de programação Java e o ambiente *Web*, o que deu interoperabilidade à ferramenta. Também foi utilizado o protocolo SNMP, permitindo a comunicação em tempo real entre a ferramenta e os equipamentos gerenciados.

A ferramenta SIGMA/WS executa funções da gerência de configuração, de desempenho e de falhas. Desse modo, auxilia na manutenção e monitoração da estrutura física da rede, de seus componentes e da sua interconectividade. Ela calcula a eficiência das atividades de comunicação e alerta para falhas e interrupções na rede.

7.3 Trabalhos Futuros

Durante o desenvolvimento deste trabalho foi possível concluir que algumas características poderiam vir a contribuir para melhorar o trabalho futuramente, por exemplo, ampliando a abrangência da ferramenta, de modo que ela possa atuar em outras áreas funcionais de gerência.

Dentre as estensões possíveis, destaca-se a contabilização e o armazenamento automático dos dados obtidos pela ferramenta, através dos quais, pode-se ampliar a

abrangência da ferramenta, atuando em outros tipos de gerência. Pois, com estes dados a ferramenta pode executar funções de Gerência Contabilidade, alarmes para a Gerência de Segurança e auxíliar na detecção de erros para a Gerência de Falhas.

Outra função que poderia ainda ser incorporada a este tipo de ferramenta seria o detalhamento do tráfego. Os dados poderiam ser analisados individualmente, em nível de VCI's ou VPI's, o que aumentaria a profundidade da análise efetuada.

Capítulo 08: Anexos

Anexo A: Código HTML da página Index.html

```
<html>

<head>
<title>SIGMA/WS</title>
</head>

<body>
<p align="center" style="word-spacing: 0; line-height: 100%; text-indent: 0; margin-
left: 0; margin-right: 0; margin-top: 8; margin-bottom: 0">
<b><font size="4">SIGMA/WS - Ferramenta para Gerência de redes ATM, via WWW, Java e
SNMP</font></b></p>
<p align="center" style="text-indent: 0; line-height: 100%; word-spacing: 0; margin-
left: 0; margin-right: 0; margin-top: 0; margin-bottom: 16"><font size="1">Sugestões e
comentários para: <ahref="mailto:douglas@inf.ufsc.br">douglas@inf.ufsc.br</a></font></p>


<div align="center">
  <center>
  <table border="0" width="752" cellspacing="0" cellpadding="0" height="467">
  <tr>
    <td width="106" valign="middle" height="312">
      <p  align="center"><a  href="redeufsc.htm"><img  border="0"  src="redeUFSC.jpg"
width="104" height="89"></a></p>
    </td>
    <td valign="middle" width="302" height="312" align="center">
      <p  align="center"  style="text-indent:  0;  line-height:  100%;  word-spacing:  0;
margin:  0"><a  href="ibm8265.htm"><img  border="0"  src="ibm8265.jpg"  width="438"
height="323" align="absmiddle"></a></p>
    </td>
    <td width="372" valign="top" height="312" colspan="2">
      <p  style="word-spacing:  0;  line-height:  100%;  text-indent:  0;  margin:  0"
align="center"><a  href="ctc.htm"><img  border="0"  src="ctc.jpg"  width="189"
height="142"></a></p>
      <p  style="word-spacing:  0;  line-height:  100%;  text-indent:  0;  margin:  0"
align="center"><a  href="cfm.htm"><img  border="0"  src="cfm.jpg"  width="189"
height="133"></a></p>
      <p  style="word-spacing:  0;  line-height:  100%;  text-indent:  0;  margin:  0"
align="center"> 
    </td>
  </tr>
  <tr>
    <td width="163" valign="top" height="155">
    </td>
    <td width="537" valign="top" height="155" colspan="2"><img border="0" src="npd1.jpg"
width="103"  height="154"><img  border="0"  src="npd2.jpg"  width="104"  height="158"><img
border="0"  src="npd3.jpg"  width="101"  height="158"><img  border="0"  src="npd4.jpg"
width="103" height="159">
    </td>
    <td width="092" valign="top" height="155"><img border="0" src="logoini.jpg">
    </td>
  </tr>
  </table>
  </center>
</div>

</body>

</html>
```

Anexo B: Código HTML da página IBM8265.html

```
<html>

<head>
<title>SIGMA/WS - ibm8265</title>
</head>

<body>

<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="15%" align="center"><b><img src="logo.jpg"></b></td>
    <td width="85%" align="center"><p align="center" style="word-spacing: 0;
line-height: 100%; text-indent: 0; margin-left: 0; margin-right: 0; margin-
top: 8; margin-bottom: 0"><b><font size="4">SIGMA/WS - Ferramenta para
Gerência de redes ATM, via WWW, Java e SNMP</font></b></p></td>
  </tr>
</table>

<p align="center"><img border="0" src="ibm8265v.jpg" width="125"
height="172"></p>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="25%" align="center"><b>Informações Gerais: </b></td>
    <td width="25%" align="center"><b>Informações sobre as Conexões</b></td>
    <td width="25%" align="center"><b>Informações sobre o Tráfego</b></td>
    <td width="25%" align="center"><b>Informações sobre a QoS</b></td> </tr>
  <tr>
    <td width="25%" align="center"><a
href="IBM8265Sistema.html">Sistema</a></td>
    <td width="25%" align="center"><a href="IBM8265LinkState.html">Estado das
Conexões</a></td>
    <td width="25%" align="center"><a href="IBM8265PacotesIP.html">Tráfego de
Pacotes IP</a></td>
    <td width="25%" align="center"><a href="IBM8265QoS.html">QoS das
Interfaces</a></td> </tr>
  <tr>
    <td width="25%" align="center"><a href="IBM8265IntFisica.html">Interfaces
</a></td>
    <td width="25%" align="center"><a href="IBM8265Trafego.html">Tráfego das
Conexões</a></td>
    <td width="25%" align="center"><a href="IBM8265PacotesTCP.html">Tráfego de
Segmentos TCP</a></td>
    <td width="25%" align="center"></td> </tr>
  <tr>
    <td width="25%" align="center"></td>
    <td width="25%" align="center"></td>
    <td width="25%" align="center"><a href="IBM8265PacotesUDP.html">Tráfego de
Datagramas UDP</a></td>
    <td width="25%" align="center"></td> </tr>
  <tr>
    <td width="25%" align="center"></td>
    <td width="25%" align="center"></td>
    <td width="25%" align="center"></td>
    <td width="25%" align="center"></td> </tr>
</table>
<p style="word-spacing: 0; line-height: 100%; text-indent: 0; margin: 0"
align="center"> </p>
<p style="word-spacing: 0; line-height: 100%; text-indent: 0; margin: 0"
align="center"> </p>
<p> </p>

</body>

</html>
```

Anexo C: Código HTML da página IBM8265Sistema.html

```
<html>

<head>
<title>SIGMA/WS - ibm8265</title>
</head>

<body>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="15%" align="center"><b><img src="logo.jpg"></b></td>
    <td width="85%" align="center"><p align="center" style="word-spacing: 0; line-
height: 100%; text-indent: 0; margin-left: 0; margin-right: 0; margin-top: 8; margin-
bottom: 0"><b><font size="4">SIGMA/WS - Ferramenta para Gerência de redes ATM, via WWW,
Java e SNMP</font></b></p></td>
  </tr>
</table>
<p align="center"><img border="0" src="ibm8265v.jpg" width="125" height="172"></p>
<p align="center"><b><font size="5">Informações sobre o Sistema</font></b></p>

<p align="center">
 <OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="760" height="239"
align="baseline"     codebase="http://java.sun.com/products/plugin/1.1/     jinstall-11-
win32.cab  #Version=1,1,0,0">
 <PARAM NAME="code" VALUE="IBM8265Sistema.class">
 <PARAM NAME="codebase" VALUE="../classes">
 <PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
 <PARAM NAME="archive"VALUE="AdventNetSnmp.jar,NetMonitor.jar">
 <COMMENT>
 <EMBED         type="application/x-java-applet"         width="760"         height="239"
code="IBM8265Sistema.class"                                          codebase="../classes"
archive="AdventNetSnmp.jar,NetMonitor.jar">
 <NOEMBED>
 </COMMENT>
 </NOEMBED>
 </EMBED>
 </OBJECT>
</p>

</body>

</html>
```

Anexo C1: Código JAVA do applet IBM8265Sistema.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class IBM8265Sistema extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(760,239);
        initialized = true;

        SnmpTextField4= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField4);
        SnmpTextField4.setBounds(100,100,650,30);

        SnmpTextField6= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField6);
        SnmpTextField6.setBounds(100,160,650,30);

        JLabel3= new javax.swing.JLabel();
        getContentPane().add(JLabel3);
        JLabel3.setBounds(10,40,90,30);

        JLabel4= new javax.swing.JLabel();
        getContentPane().add(JLabel4);
        JLabel4.setBounds(10,70,90,30);

        JLabel5= new javax.swing.JLabel();
        getContentPane().add(JLabel5);
        JLabel5.setBounds(10,100,90,30);

        JLabel6= new javax.swing.JLabel();
        getContentPane().add(JLabel6);
        JLabel6.setBounds(10,130,90,30);

        JLabel7= new javax.swing.JLabel();
        getContentPane().add(JLabel7);
        JLabel7.setBounds(10,160,90,30);

        JLabel8= new javax.swing.JLabel();
        getContentPane().add(JLabel8);
        JLabel8.setBounds(10,190,90,30);

        JLabel2= new javax.swing.JLabel();
        getContentPane().add(JLabel2);
        JLabel2.setBounds(10,10,90,30);

        SnmpTextField1= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField1);
        SnmpTextField1.setBounds(100,10,650,30);

        SnmpTextField2= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField2);
        SnmpTextField2.setBounds(100,40,650,30);

        SnmpTextField5= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField5);
        SnmpTextField5.setBounds(100,130,650,30);

        SnmpTextField7= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField7);
        SnmpTextField7.setBounds(100,190,650,30);

        SnmpTextField3= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField3);
        SnmpTextField3.setBounds(100,70,650,30);

       try
```

```
{
    SnmpTextField4.setTargetHost("150.162.252.20");
    SnmpTextField4.setCommunity("painel");
    SnmpTextField4.setObjectID(".1.3.6.1.2.1.1.4.0");
    SnmpTextField4.setNonRepeaters(1);
    SnmpTextField4.setPollInterval(300);
    SnmpTextField4.setBackground(new Color(-1));
    SnmpTextField4.setForeground(new Color(-16777216));
}catch(Exception ex){};

try
{
    SnmpTextField6.setTargetHost("150.162.252.20");
    SnmpTextField6.setCommunity("painel");
    SnmpTextField6.setObjectID(".1.3.6.1.2.1.1.6.0");
    SnmpTextField6.setPollInterval(300);
    SnmpTextField6.setBackground(new Color(-1));
    SnmpTextField6.setForeground(new Color(-16777216));
}catch(Exception ex){};

try
{
    JLabel3.setText("sysObjectID");
}catch(Exception ex){};

try
{
    JLabel4.setText("sysUpTime");
}catch(Exception ex){};

try
{
    JLabel5.setText("sysContact");
}catch(Exception ex){};

try
{
    JLabel6.setText("sysName");
}catch(Exception ex){};

try
{
    JLabel7.setText("sysLocation");
}catch(Exception ex){};

try
{
    JLabel8.setText("sysServices");
}catch(Exception ex){};

try
{
    JLabel2.setText("sysDescr");
}catch(Exception ex){};

try
{
    SnmpTextField1.setTargetHost("150.162.252.20");
    SnmpTextField1.setCommunity("painel");
    SnmpTextField1.setObjectID(".1.3.6.1.2.1.1.1.0");
    SnmpTextField1.setNonRepeaters(1);
    SnmpTextField1.setPollInterval(300);
    SnmpTextField1.setBackground(new Color(-1));
    SnmpTextField1.setForeground(new Color(-16777216));
}catch(Exception ex){};

try
{
    SnmpTextField2.setTargetHost("150.162.252.20");
    SnmpTextField2.setCommunity("painel");
    SnmpTextField2.setObjectID(".1.3.6.1.2.1.1.2.0");
    SnmpTextField2.setBackground(new Color(-1));
    SnmpTextField2.setForeground(new Color(-16777216));
    SnmpTextField2.setPollInterval(10);
}catch(Exception ex){};

try
{
```

```
            SnmpTextField5.setTargetHost("150.162.252.20");
            SnmpTextField5.setCommunity("painel");
            SnmpTextField5.setObjectID(".1.3.6.1.2.1.1.5.0");
            SnmpTextField5.setPollInterval(300);
            SnmpTextField5.setNonRepeaters(0);
            SnmpTextField5.setBackground(new Color(-1));
            SnmpTextField5.setForeground(new Color(-16777216));
        }catch(Exception ex){};

        try
        {
            SnmpTextField7.setTargetHost("150.162.252.20");
            SnmpTextField7.setCommunity("painel");
            SnmpTextField7.setObjectID(".1.3.6.1.2.1.1.7.0");
            SnmpTextField7.setPollInterval(300);
            SnmpTextField7.setBackground(new Color(-1));
            SnmpTextField7.setForeground(new Color(-16777216));
        }catch(Exception ex){};

        try
        {
            SnmpTextField3.setTargetHost("150.162.252.20");
            SnmpTextField3.setCommunity("painel");
            SnmpTextField3.setObjectID(".1.3.6.1.2.1.1.3.0");
            SnmpTextField3.setBackground(new Color(-1));
            SnmpTextField3.setForeground(new Color(-16777216));
            SnmpTextField3.setPollInterval(5);
        }catch(Exception ex){};
    }

    com.adventnet.netmonitor.SnmpTextField  SnmpTextField4 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField6 = null;
    javax.swing.JLabel  JLabel3 = null;
    javax.swing.JLabel  JLabel4 = null;
    javax.swing.JLabel  JLabel5 = null;
    javax.swing.JLabel  JLabel6 = null;
    javax.swing.JLabel  JLabel7 = null;
    javax.swing.JLabel  JLabel8 = null;
    javax.swing.JLabel  JLabel2 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField1 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField2 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField5 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField7 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField3 = null;
}
```

Anexo D: Código HTML da página IBM8265IntFisica.html

```
<html>

<head>
<meta http-equiv="Content-Language" content="pt-br">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>SIGMA/WS - ibm8265</title>
</head>

<body>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="15%" align="center"><b><img src="logo.jpg"></b></td>
    <td width="85%" align="center"><p align="center" style="word-spacing: 0; line-
height: 100%; text-indent: 0; margin-left: 0; margin-right: 0; margin-top: 8; margin-
bottom: 0"><b><font size="4">SIGMA/WS - Ferramenta para Gerência de redes ATM, via WWW,
Java e SNMP</font></b></p></td>
  </tr>
</table>
<p align="center"><img border="0" src="ibm8265v.jpg" width="125" height="172"></p>
<p align="center"><b><font size="5">Informações sobre as Interfaces
Físicas</font></b></p>

<p align="center">
 <OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="692" height="586"
align="baseline" codebase="http://java.sun.com/products/plugin/1.1/ jinstall-11-
win32.cab #Version=1,1,0,0">
 <PARAM NAME="code" VALUE="IBM8265IntFisica.class">
 <PARAM NAME="codebase" VALUE="../classes">
 <PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
 <PARAM
NAME="archive"VALUE="BuilderSwing.jar,AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
 <COMMENT>
 <EMBED type="application/x-java-applet" width="692" height="586"
code="IBM8265IntFisica.class" codebase="../classes"
archive="BuilderSwing.jar,AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
 <NOEMBED>
 </COMMENT>
 </NOEMBED>
 </EMBED>
 </OBJECT>
</p>

</body>

</html>
```

Anexo D1: Código JAVA do applet IBM8265IntFisica.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class IBM8265IntFisica extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(692,586);
        initialized = true;

        JLabel1= new javax.swing.JLabel();
        getContentPane().add(JLabel1);
        JLabel1.setBounds(10,15,30,20);

        SnmpTextField1= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField1);
        SnmpTextField1.setBounds(10,35,30,30);

        JLabel4= new javax.swing.JLabel();
        getContentPane().add(JLabel4);
        JLabel4.setBounds(40,15,30,20);

        SnmpTextField4= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField4);
        SnmpTextField4.setBounds(40,35,30,30);

        JLabel5= new javax.swing.JLabel();
        getContentPane().add(JLabel5);
        JLabel5.setBounds(70,15,40,20);

        SnmpTextField5= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField5);
        SnmpTextField5.setBounds(70,35,40,30);

        JLabel7= new javax.swing.JLabel();
        getContentPane().add(JLabel7);
        JLabel7.setBounds(340,15,140,20);

        SnmpLed2= new com.adventnet.netmonitor.SnmpLed(this);
        SnmpLed2.setLayout(null);
        getContentPane().add(SnmpLed2);
        SnmpLed2.setBounds(340,35,140,30);

        SnmpLed3= new com.adventnet.netmonitor.SnmpLed(this);
        SnmpLed3.setLayout(null);
        getContentPane().add(SnmpLed3);
        SnmpLed3.setBounds(110,35,120,30);

        JLabel2= new javax.swing.JLabel();
        getContentPane().add(JLabel2);
        JLabel2.setBounds(110,15,120,20);

        JLabel6= new javax.swing.JLabel();
        getContentPane().add(JLabel6);
        JLabel6.setBounds(230,15,110,20);

        SnmpLed1= new com.adventnet.netmonitor.SnmpLed(this);
        SnmpLed1.setLayout(null);
        getContentPane().add(SnmpLed1);
        SnmpLed1.setBounds(230,35,110,30);

        SnmpTextField6= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField6);
        SnmpTextField6.setBounds(40,65,30,30);

        SnmpTextField9= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField9);
```

```
SnmpTextField9.setBounds(10,95,30,30);

SnmpTextField10= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField10);
SnmpTextField10.setBounds(40,95,30,30);

SnmpTextField7= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField7);
SnmpTextField7.setBounds(70,65,40,30);

SnmpLed6= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed6.setLayout(null);
getContentPane().add(SnmpLed6);
SnmpLed6.setBounds(110,65,120,30);

SnmpLed7= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed7.setLayout(null);
getContentPane().add(SnmpLed7);
SnmpLed7.setBounds(230,65,110,30);

SnmpLed13= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed13.setLayout(null);
getContentPane().add(SnmpLed13);
SnmpLed13.setBounds(340,95,140,30);

SnmpLed8= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed8.setLayout(null);
getContentPane().add(SnmpLed8);
SnmpLed8.setBounds(340,65,140,30);

SnmpLed11= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed11.setLayout(null);
getContentPane().add(SnmpLed11);
SnmpLed11.setBounds(110,95,120,30);

SnmpLed12= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed12.setLayout(null);
getContentPane().add(SnmpLed12);
SnmpLed12.setBounds(230,95,110,30);

SnmpTextField11= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField11);
SnmpTextField11.setBounds(70,95,40,30);

SnmpTextField13= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField13);
SnmpTextField13.setBounds(70,125,40,30);

SnmpTextField15= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField15);
SnmpTextField15.setBounds(70,185,40,30);

SnmpTextField19= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField19);
SnmpTextField19.setBounds(70,215,40,30);

SnmpTextField22= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField22);
SnmpTextField22.setBounds(70,275,40,30);

SnmpTextField23= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField23);
SnmpTextField23.setBounds(70,305,40,30);

SnmpTextField26= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField26);
SnmpTextField26.setBounds(70,335,40,30);

SnmpTextField16= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField16);
SnmpTextField16.setBounds(70,365,40,30);

SnmpTextField17= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField17);
SnmpTextField17.setBounds(70,395,40,30);

SnmpTextField14= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField14);
```

```
SnmpTextField14.setBounds(70,155,40,30);

SnmpTextField29= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField29);
SnmpTextField29.setBounds(10,155,30,30);

SnmpTextField28= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField28);
SnmpTextField28.setBounds(10,125,30,30);

SnmpTextField30= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField30);
SnmpTextField30.setBounds(10,185,30,30);

SnmpTextField31= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField31);
SnmpTextField31.setBounds(10,215,30,30);

SnmpTextField32= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField32);
SnmpTextField32.setBounds(10,245,30,30);

SnmpTextField33= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField33);
SnmpTextField33.setBounds(10,275,30,30);

SnmpTextField34= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField34);
SnmpTextField34.setBounds(10,305,30,30);

SnmpTextField35= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField35);
SnmpTextField35.setBounds(10,335,30,30);

SnmpTextField36= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField36);
SnmpTextField36.setBounds(10,365,30,30);

SnmpTextField38= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField38);
SnmpTextField38.setBounds(10,425,30,30);

SnmpTextField18= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField18);
SnmpTextField18.setBounds(70,545,40,30);

SnmpTextField25= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField25);
SnmpTextField25.setBounds(70,515,40,30);

SnmpTextField24= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField24);
SnmpTextField24.setBounds(70,485,40,30);

SnmpTextField27= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField27);
SnmpTextField27.setBounds(70,455,40,30);

SnmpTextField21= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField21);
SnmpTextField21.setBounds(70,425,40,30);

SnmpTextField40= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField40);
SnmpTextField40.setBounds(10,485,30,30);

SnmpTextField39= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField39);
SnmpTextField39.setBounds(10,455,30,30);

SnmpTextField41= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField41);
SnmpTextField41.setBounds(10,515,30,30);

SnmpTextField42= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField42);
SnmpTextField42.setBounds(10,545,30,30);
```

```
SnmpTextField43= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField43);
SnmpTextField43.setBounds(40,125,30,30);

SnmpTextField45= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField45);
SnmpTextField45.setBounds(40,185,30,30);

SnmpTextField46= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField46);
SnmpTextField46.setBounds(40,215,30,30);

SnmpTextField47= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField47);
SnmpTextField47.setBounds(40,245,30,30);

SnmpTextField48= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField48);
SnmpTextField48.setBounds(40,275,30,30);

SnmpTextField49= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField49);
SnmpTextField49.setBounds(40,305,30,30);

SnmpTextField52= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField52);
SnmpTextField52.setBounds(40,365,30,30);

SnmpTextField53= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField53);
SnmpTextField53.setBounds(40,395,30,30);

SnmpTextField54= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField54);
SnmpTextField54.setBounds(40,425,30,30);

SnmpTextField55= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField55);
SnmpTextField55.setBounds(40,455,30,30);

SnmpTextField57= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField57);
SnmpTextField57.setBounds(40,515,30,30);

SnmpTextField58= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField58);
SnmpTextField58.setBounds(40,545,30,30);

SnmpLed16= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed16.setLayout(null);
getContentPane().add(SnmpLed16);
SnmpLed16.setBounds(110,125,120,30);

SnmpLed17= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed17.setLayout(null);
getContentPane().add(SnmpLed17);
SnmpLed17.setBounds(110,155,120,30);

SnmpLed18= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed18.setLayout(null);
getContentPane().add(SnmpLed18);
SnmpLed18.setBounds(110,185,120,30);

SnmpLed19= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed19.setLayout(null);
getContentPane().add(SnmpLed19);
SnmpLed19.setBounds(110,215,120,30);

SnmpLed20= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed20.setLayout(null);
getContentPane().add(SnmpLed20);
SnmpLed20.setBounds(110,245,120,30);

SnmpLed21= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed21.setLayout(null);
getContentPane().add(SnmpLed21);
SnmpLed21.setBounds(110,275,120,30);
```

```
SnmpLed22= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed22.setLayout(null);
getContentPane().add(SnmpLed22);
SnmpLed22.setBounds(110,305,120,30);

SnmpLed23= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed23.setLayout(null);
getContentPane().add(SnmpLed23);
SnmpLed23.setBounds(110,335,120,30);

SnmpLed25= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed25.setLayout(null);
getContentPane().add(SnmpLed25);
SnmpLed25.setBounds(110,395,120,30);

SnmpLed27= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed27.setLayout(null);
getContentPane().add(SnmpLed27);
SnmpLed27.setBounds(110,455,120,30);

SnmpLed28= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed28.setLayout(null);
getContentPane().add(SnmpLed28);
SnmpLed28.setBounds(110,485,120,30);

SnmpLed29= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed29.setLayout(null);
getContentPane().add(SnmpLed29);
SnmpLed29.setBounds(110,515,120,30);

SnmpLed30= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed30.setLayout(null);
getContentPane().add(SnmpLed30);
SnmpLed30.setBounds(110,545,120,30);

SnmpLed31= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed31.setLayout(null);
getContentPane().add(SnmpLed31);
SnmpLed31.setBounds(230,125,110,30);

SnmpLed32= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed32.setLayout(null);
getContentPane().add(SnmpLed32);
SnmpLed32.setBounds(230,155,110,30);

SnmpLed33= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed33.setLayout(null);
getContentPane().add(SnmpLed33);
SnmpLed33.setBounds(230,185,110,30);

SnmpLed34= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed34.setLayout(null);
getContentPane().add(SnmpLed34);
SnmpLed34.setBounds(230,215,110,30);

SnmpLed35= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed35.setLayout(null);
getContentPane().add(SnmpLed35);
SnmpLed35.setBounds(230,245,110,30);

SnmpLed37= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed37.setLayout(null);
getContentPane().add(SnmpLed37);
SnmpLed37.setBounds(230,305,110,30);

SnmpLed38= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed38.setLayout(null);
getContentPane().add(SnmpLed38);
SnmpLed38.setBounds(230,335,110,30);

SnmpLed39= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed39.setLayout(null);
getContentPane().add(SnmpLed39);
SnmpLed39.setBounds(230,365,110,30);

SnmpLed55= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed55.setLayout(null);
SnmpLed39.add(SnmpLed55);
```

```
SnmpLed55.setBounds(105,25,140,30);

SnmpLed40= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed40.setLayout(null);
getContentPane().add(SnmpLed40);
SnmpLed40.setBounds(230,395,110,30);

SnmpLed41= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed41.setLayout(null);
getContentPane().add(SnmpLed41);
SnmpLed41.setBounds(230,425,110,30);

SnmpLed42= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed42.setLayout(null);
getContentPane().add(SnmpLed42);
SnmpLed42.setBounds(230,455,110,30);

SnmpLed43= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed43.setLayout(null);
getContentPane().add(SnmpLed43);
SnmpLed43.setBounds(230,485,110,30);

SnmpLed44= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed44.setLayout(null);
getContentPane().add(SnmpLed44);
SnmpLed44.setBounds(230,515,110,30);

SnmpLed45= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed45.setLayout(null);
getContentPane().add(SnmpLed45);
SnmpLed45.setBounds(230,545,110,30);

SnmpLed46= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed46.setLayout(null);
getContentPane().add(SnmpLed46);
SnmpLed46.setBounds(340,125,140,30);

SnmpLed47= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed47.setLayout(null);
getContentPane().add(SnmpLed47);
SnmpLed47.setBounds(340,155,140,30);

SnmpLed48= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed48.setLayout(null);
getContentPane().add(SnmpLed48);
SnmpLed48.setBounds(340,185,140,30);

SnmpLed50= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed50.setLayout(null);
getContentPane().add(SnmpLed50);
SnmpLed50.setBounds(340,245,140,30);

SnmpLed51= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed51.setLayout(null);
getContentPane().add(SnmpLed51);
SnmpLed51.setBounds(340,275,140,30);

SnmpLed52= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed52.setLayout(null);
getContentPane().add(SnmpLed52);
SnmpLed52.setBounds(340,305,140,30);

SnmpLed53= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed53.setLayout(null);
getContentPane().add(SnmpLed53);
SnmpLed53.setBounds(340,335,140,30);

SnmpLed56= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed56.setLayout(null);
getContentPane().add(SnmpLed56);
SnmpLed56.setBounds(340,425,140,30);

SnmpLed57= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed57.setLayout(null);
getContentPane().add(SnmpLed57);
SnmpLed57.setBounds(340,455,140,30);

SnmpLed58= new com.adventnet.netmonitor.SnmpLed(this);
```

```
SnmpLed58.setLayout(null);
getContentPane().add(SnmpLed58);
SnmpLed58.setBounds(340,485,140,30);

SnmpLed60= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed60.setLayout(null);
getContentPane().add(SnmpLed60);
SnmpLed60.setBounds(340,545,140,30);

SnmpTextField2= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField2);
SnmpTextField2.setBounds(10,65,30,30);

SnmpTextField44= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField44);
SnmpTextField44.setBounds(40,155,30,30);

SnmpTextField20= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField20);
SnmpTextField20.setBounds(70,245,40,30);

SnmpTextField51= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField51);
SnmpTextField51.setBounds(40,335,30,30);

SnmpTextField37= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField37);
SnmpTextField37.setBounds(10,395,30,30);

SnmpTextField56= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField56);
SnmpTextField56.setBounds(40,485,30,30);

SnmpLed24= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed24.setLayout(null);
getContentPane().add(SnmpLed24);
SnmpLed24.setBounds(110,365,120,30);

SnmpLed26= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed26.setLayout(null);
getContentPane().add(SnmpLed26);
SnmpLed26.setBounds(110,425,120,30);

SnmpLed36= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed36.setLayout(null);
getContentPane().add(SnmpLed36);
SnmpLed36.setBounds(230,275,110,30);

SnmpLed49= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed49.setLayout(null);
getContentPane().add(SnmpLed49);
SnmpLed49.setBounds(340,215,140,30);

SnmpLed59= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed59.setLayout(null);
getContentPane().add(SnmpLed59);
SnmpLed59.setBounds(340,515,140,30);

JLabel8= new javax.swing.JLabel();
getContentPane().add(JLabel8);
JLabel8.setBounds(480,15,120,20);

SnmpLed5= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed5.setLayout(null);
getContentPane().add(SnmpLed5);
SnmpLed5.setBounds(480,35,120,30);

SnmpLed10= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed10.setLayout(null);
getContentPane().add(SnmpLed10);
SnmpLed10.setBounds(480,65,120,30);

SnmpLed15= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed15.setLayout(null);
getContentPane().add(SnmpLed15);
SnmpLed15.setBounds(480,95,120,30);

SnmpLed76= new com.adventnet.netmonitor.SnmpLed(this);
```

```
SnmpLed76.setLayout(null);
getContentPane().add(SnmpLed76);
SnmpLed76.setBounds(480,125,120,30);

SnmpLed77= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed77.setLayout(null);
getContentPane().add(SnmpLed77);
SnmpLed77.setBounds(480,155,120,30);

SnmpLed78= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed78.setLayout(null);
getContentPane().add(SnmpLed78);
SnmpLed78.setBounds(480,185,120,30);

SnmpLed82= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed82.setLayout(null);
getContentPane().add(SnmpLed82);
SnmpLed82.setBounds(480,215,120,30);

SnmpLed80= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed80.setLayout(null);
getContentPane().add(SnmpLed80);
SnmpLed80.setBounds(480,245,120,30);

SnmpLed81= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed81.setLayout(null);
getContentPane().add(SnmpLed81);
SnmpLed81.setBounds(480,275,120,30);

SnmpLed83= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed83.setLayout(null);
getContentPane().add(SnmpLed83);
SnmpLed83.setBounds(480,305,120,30);

SnmpLed84= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed84.setLayout(null);
getContentPane().add(SnmpLed84);
SnmpLed84.setBounds(480,335,120,30);

SnmpLed91= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed91.setLayout(null);
getContentPane().add(SnmpLed91);
SnmpLed91.setBounds(480,365,120,30);

SnmpLed85= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed85.setLayout(null);
getContentPane().add(SnmpLed85);
SnmpLed85.setBounds(480,395,120,30);

SnmpLed86= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed86.setLayout(null);
getContentPane().add(SnmpLed86);
SnmpLed86.setBounds(480,425,120,30);

SnmpLed87= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed87.setLayout(null);
getContentPane().add(SnmpLed87);
SnmpLed87.setBounds(480,455,120,30);

SnmpLed88= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed88.setLayout(null);
getContentPane().add(SnmpLed88);
SnmpLed88.setBounds(480,485,120,30);

SnmpLed89= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed89.setLayout(null);
getContentPane().add(SnmpLed89);
SnmpLed89.setBounds(480,515,120,30);

SnmpLed90= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed90.setLayout(null);
getContentPane().add(SnmpLed90);
SnmpLed90.setBounds(480,545,120,30);

JLabel3= new javax.swing.JLabel();
getContentPane().add(JLabel3);
JLabel3.setBounds(600,15,80,20);
```

```
SnmpTextField3= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField3);
SnmpTextField3.setBounds(600,35,80,30);

SnmpTextField12= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField12);
SnmpTextField12.setBounds(600,95,80,30);

SnmpTextField60= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField60);
SnmpTextField60.setBounds(600,125,80,30);

SnmpTextField61= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField61);
SnmpTextField61.setBounds(600,155,80,30);

SnmpTextField64= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField64);
SnmpTextField64.setBounds(600,245,80,30);

SnmpTextField66= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField66);
SnmpTextField66.setBounds(600,275,80,30);

SnmpTextField67= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField67);
SnmpTextField67.setBounds(600,305,80,30);

SnmpTextField68= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField68);
SnmpTextField68.setBounds(600,335,80,30);

SnmpTextField69= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField69);
SnmpTextField69.setBounds(600,365,80,30);

SnmpTextField71= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField71);
SnmpTextField71.setBounds(600,425,80,30);

SnmpTextField72= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField72);
SnmpTextField72.setBounds(600,455,80,30);

SnmpTextField73= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField73);
SnmpTextField73.setBounds(600,485,80,30);

SnmpTextField65= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField65);
SnmpTextField65.setBounds(735,350,80,30);

SnmpLed54= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed54.setLayout(null);
getContentPane().add(SnmpLed54);
SnmpLed54.setBounds(340,365,140,30);

SnmpLed92= new com.adventnet.netmonitor.SnmpLed(this);
SnmpLed92.setLayout(null);
getContentPane().add(SnmpLed92);
SnmpLed92.setBounds(340,395,140,30);

SnmpTextField74= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField74);
SnmpTextField74.setBounds(600,515,80,30);

SnmpTextField76= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField76);
SnmpTextField76.setBounds(600,545,80,30);

SnmpTextField70= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField70);
SnmpTextField70.setBounds(600,395,80,30);

SnmpTextField63= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField63);
SnmpTextField63.setBounds(600,215,80,30);
```

```
SnmpTextField62= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField62);
SnmpTextField62.setBounds(600,185,80,30);

SnmpTextField8= new com.adventnet.netmonitor.SnmpTextField(this);
getContentPane().add(SnmpTextField8);
SnmpTextField8.setBounds(600,65,80,30);

try
{
    JLabel1.setText("Slot");
}catch(Exception ex){};

try
{
    SnmpTextField1.setCommunity("painel");
    SnmpTextField1.setTargetHost("150.162.252.20");
    SnmpTextField1.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.201");
    SnmpTextField1.setPollInterval(300);
}catch(Exception ex){};

try
{
    JLabel4.setText("Port");
}catch(Exception ex){};

try
{
    SnmpTextField4.setCommunity("painel");
    SnmpTextField4.setTargetHost("150.162.252.20");
    SnmpTextField4.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.201");
    SnmpTextField4.setPollInterval(300);
}catch(Exception ex){};

try
{
    JLabel5.setText("Index");
}catch(Exception ex){};

try
{
    SnmpTextField5.setCommunity("painel");
    SnmpTextField5.setTargetHost("150.162.252.20");
    SnmpTextField5.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.201");
    SnmpTextField5.setPollInterval(300);
}catch(Exception ex){};

try
{
    JLabel7.setText("MediaType");
}catch(Exception ex){};

try
{
    SnmpLed2.setShowValue(false);
    SnmpLed2.setThresholdValue1(1);
    SnmpLed2.setThresholdValue2(2);
    SnmpLed2.setThresholdValue3(3);
    SnmpLed2.setCommunity("painel");
    SnmpLed2.setTargetPort(161);
    SnmpLed2.setTargetHost("150.162.252.20");
    SnmpLed2.setThresholdLabel1("Unknow");
    SnmpLed2.setThresholdValue4(4);
    SnmpLed2.setThresholdValue5(5);
    SnmpLed2.setThresholdValue6(6);
    SnmpLed2.setThresholdValue7(7);
    SnmpLed2.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.201");
    SnmpLed2.setThresholdLabel2("Monomode-Fiber");
    SnmpLed2.setThresholdLabel3("Multimode-Fiber");
    SnmpLed2.setThresholdLabel4("Twisted-Pair");
    SnmpLed2.setThresholdLabel5("UTP");
    SnmpLed2.setThresholdLabel6("STP");
    SnmpLed2.setThresholdLabel7("Coaxial Cable");
    SnmpLed2.setNumberOfStates(7);
    SnmpLed2.setPollInterval(300);
    SnmpLed2.setBgColor(new Color(-1));
    SnmpLed2.setFgColor(new Color(-16777216));
    SnmpLed2.setThresholdColor1(new Color(-1));
```

```
    SnmpLed2.setThresholdColor2(new Color(-1));
    SnmpLed2.setThresholdColor3(new Color(-1));
    SnmpLed2.setThresholdColor4(new Color(-1));
    SnmpLed2.setThresholdColor5(new Color(-1));
    SnmpLed2.setThresholdColor6(new Color(-1));
    SnmpLed2.setThresholdColor7(new Color(-1));
}catch(Exception ex){};

try
{
    SnmpLed3.setShowValue(false);
    SnmpLed3.setThresholdValue1(1);
    SnmpLed3.setThresholdValue2(2);
    SnmpLed3.setThresholdValue3(3);
    SnmpLed3.setCommunity("painel");
    SnmpLed3.setTargetPort(161);
    SnmpLed3.setTargetHost("150.162.252.20");
    SnmpLed3.setThresholdLabel1("Unknow");
    SnmpLed3.setThresholdValue4(4);
    SnmpLed3.setThresholdValue5(5);
    SnmpLed3.setThresholdValue6(6);
    SnmpLed3.setThresholdValue7(7);
    SnmpLed3.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.201");
    SnmpLed3.setThresholdLabel2("Private UNI");
    SnmpLed3.setThresholdLabel3("Private NNI");
    SnmpLed3.setThresholdLabel4("Public UNI");
    SnmpLed3.setThresholdLabel5("GSMP");
    SnmpLed3.setThresholdLabel6("Void");
    SnmpLed3.setThresholdLabel7("Auto");
    SnmpLed3.setNumberOfStates(7);
    SnmpLed3.setPollInterval(1);
    SnmpLed3.setBgColor(new Color(-1));
    SnmpLed3.setFgColor(new Color(-16777216));
    SnmpLed3.setThresholdColor1(new Color(-1));
    SnmpLed3.setThresholdColor2(new Color(-1));
    SnmpLed3.setThresholdColor3(new Color(-1));
    SnmpLed3.setThresholdColor4(new Color(-1));
    SnmpLed3.setThresholdColor5(new Color(-1));
    SnmpLed3.setThresholdColor6(new Color(-1));
    SnmpLed3.setThresholdColor7(new Color(-1));
}catch(Exception ex){};

try
{
    JLabel2.setText("ATMAccess");
}catch(Exception ex){};

try
{
    JLabel6.setText("Connector");
}catch(Exception ex){};

try
{
    SnmpLed1.setShowValue(false);
    SnmpLed1.setThresholdValue1(1);
    SnmpLed1.setThresholdValue2(2);
    SnmpLed1.setThresholdValue3(3);
    SnmpLed1.setCommunity("painel");
    SnmpLed1.setTargetPort(161);
    SnmpLed1.setTargetHost("150.162.252.20");
    SnmpLed1.setThresholdLabel1("Unknow");
    SnmpLed1.setThresholdValue4(4);
    SnmpLed1.setThresholdValue5(5);
    SnmpLed1.setThresholdValue6(6);
    SnmpLed1.setThresholdValue7(7);
    SnmpLed1.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.201");
    SnmpLed1.setThresholdLabel3("Mic");
    SnmpLed1.setThresholdLabel2("Internal");
    SnmpLed1.setThresholdLabel4("SC-Duplex");
    SnmpLed1.setThresholdLabel5("Monomode");
    SnmpLed1.setThresholdLabel6("DB-9");
    SnmpLed1.setThresholdLabel7("RJ-45");
    SnmpLed1.setNumberOfStates(7);
    SnmpLed1.setPollInterval(300);
    SnmpLed1.setBgColor(new Color(-1));
    SnmpLed1.setFgColor(new Color(-16711680));
    SnmpLed1.setThresholdColor1(new Color(-1));
```

```
        SnmpLed1.setThresholdColor2(new Color(-1));
        SnmpLed1.setThresholdColor3(new Color(-1));
        SnmpLed1.setThresholdColor4(new Color(-1));
        SnmpLed1.setThresholdColor5(new Color(-1));
        SnmpLed1.setThresholdColor6(new Color(-1));
        SnmpLed1.setThresholdColor7(new Color(-1));
    }catch(Exception ex){};

    try
    {
        SnmpTextField6.setCommunity("painel");
        SnmpTextField6.setTargetHost("150.162.252.20");
        SnmpTextField6.setPollInterval(300);
        SnmpTextField6.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.202");
    }catch(Exception ex){};

    try
    {
        SnmpTextField9.setCommunity("painel");
        SnmpTextField9.setTargetHost("150.162.252.20");
        SnmpTextField9.setPollInterval(300);
        SnmpTextField9.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.203");
    }catch(Exception ex){};

    try
    {
        SnmpTextField10.setCommunity("painel");
        SnmpTextField10.setTargetHost("150.162.252.20");
        SnmpTextField10.setPollInterval(300);
        SnmpTextField10.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.203");
    }catch(Exception ex){};

    try
    {
        SnmpTextField7.setCommunity("painel");
        SnmpTextField7.setTargetHost("150.162.252.20");
        SnmpTextField7.setPollInterval(300);
        SnmpTextField7.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.202");
    }catch(Exception ex){};

    try
    {
        SnmpLed6.setShowValue(false);
        SnmpLed6.setThresholdValue1(1);
        SnmpLed6.setThresholdValue2(2);
        SnmpLed6.setThresholdValue3(3);
        SnmpLed6.setCommunity("painel");
        SnmpLed6.setTargetPort(161);
        SnmpLed6.setTargetHost("150.162.252.20");
        SnmpLed6.setThresholdLabel1("Unknow");
        SnmpLed6.setThresholdValue4(4);
        SnmpLed6.setThresholdValue5(5);
        SnmpLed6.setThresholdValue6(6);
        SnmpLed6.setThresholdValue7(7);
        SnmpLed6.setThresholdLabel2("Private UNI");
        SnmpLed6.setThresholdLabel3("Private NNI");
        SnmpLed6.setThresholdLabel4("Public UNI");
        SnmpLed6.setThresholdLabel5("GSMP");
        SnmpLed6.setThresholdLabel6("Void");
        SnmpLed6.setThresholdLabel7("Auto");
        SnmpLed6.setNumberOfStates(7);
        SnmpLed6.setPollInterval(1);
        SnmpLed6.setBgColor(new Color(-1));
        SnmpLed6.setFgColor(new Color(-16777216));
        SnmpLed6.setThresholdColor1(new Color(-1));
        SnmpLed6.setThresholdColor2(new Color(-1));
        SnmpLed6.setThresholdColor3(new Color(-1));
        SnmpLed6.setThresholdColor4(new Color(-1));
        SnmpLed6.setThresholdColor5(new Color(-1));
        SnmpLed6.setThresholdColor6(new Color(-1));
        SnmpLed6.setThresholdColor7(new Color(-1));
        SnmpLed6.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.202");
    }catch(Exception ex){};

    try
    {
        SnmpLed7.setShowValue(false);
        SnmpLed7.setThresholdValue1(1);
```

```
    SnmpLed7.setThresholdValue2(2);
    SnmpLed7.setThresholdValue3(3);
    SnmpLed7.setCommunity("painel");
    SnmpLed7.setTargetPort(161);
    SnmpLed7.setTargetHost("150.162.252.20");
    SnmpLed7.setThresholdLabel1("Unknow");
    SnmpLed7.setThresholdValue4(4);
    SnmpLed7.setThresholdValue5(5);
    SnmpLed7.setThresholdValue6(6);
    SnmpLed7.setThresholdValue7(7);
    SnmpLed7.setThresholdLabel3("Mic");
    SnmpLed7.setThresholdLabel2("Internal");
    SnmpLed7.setThresholdLabel4("SC-Duplex");
    SnmpLed7.setThresholdLabel5("Monomode");
    SnmpLed7.setThresholdLabel6("DB-9");
    SnmpLed7.setThresholdLabel7("RJ-45");
    SnmpLed7.setNumberOfStates(7);
    SnmpLed7.setPollInterval(300);
    SnmpLed7.setBgColor(new Color(-1));
    SnmpLed7.setFgColor(new Color(-16711680));
    SnmpLed7.setThresholdColor1(new Color(-1));
    SnmpLed7.setThresholdColor2(new Color(-1));
    SnmpLed7.setThresholdColor3(new Color(-1));
    SnmpLed7.setThresholdColor4(new Color(-1));
    SnmpLed7.setThresholdColor5(new Color(-1));
    SnmpLed7.setThresholdColor6(new Color(-1));
    SnmpLed7.setThresholdColor7(new Color(-1));
    SnmpLed7.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.202");
}catch(Exception ex){};

try
{
    SnmpLed13.setShowValue(false);
    SnmpLed13.setThresholdValue1(1);
    SnmpLed13.setThresholdValue2(2);
    SnmpLed13.setThresholdValue3(3);
    SnmpLed13.setCommunity("painel");
    SnmpLed13.setTargetPort(161);
    SnmpLed13.setTargetHost("150.162.252.20");
    SnmpLed13.setThresholdLabel1("Unknow");
    SnmpLed13.setThresholdValue4(4);
    SnmpLed13.setThresholdValue5(5);
    SnmpLed13.setThresholdValue6(6);
    SnmpLed13.setThresholdValue7(7);
    SnmpLed13.setThresholdLabel2("Monomode-Fiber");
    SnmpLed13.setThresholdLabel3("Multimode-Fiber");
    SnmpLed13.setThresholdLabel4("Twisted-Pair");
    SnmpLed13.setThresholdLabel5("UTP");
    SnmpLed13.setThresholdLabel6("STP");
    SnmpLed13.setThresholdLabel7("Coaxial Cable");
    SnmpLed13.setNumberOfStates(7);
    SnmpLed13.setPollInterval(300);
    SnmpLed13.setBgColor(new Color(-1));
    SnmpLed13.setFgColor(new Color(-16777216));
    SnmpLed13.setThresholdColor1(new Color(-1));
    SnmpLed13.setThresholdColor2(new Color(-1));
    SnmpLed13.setThresholdColor3(new Color(-1));
    SnmpLed13.setThresholdColor4(new Color(-1));
    SnmpLed13.setThresholdColor5(new Color(-1));
    SnmpLed13.setThresholdColor6(new Color(-1));
    SnmpLed13.setThresholdColor7(new Color(-1));
    SnmpLed13.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.203");
}catch(Exception ex){};

try
{
    SnmpLed8.setShowValue(false);
    SnmpLed8.setThresholdValue1(1);
    SnmpLed8.setThresholdValue2(2);
    SnmpLed8.setThresholdValue3(3);
    SnmpLed8.setCommunity("painel");
    SnmpLed8.setTargetPort(161);
    SnmpLed8.setTargetHost("150.162.252.20");
    SnmpLed8.setThresholdLabel1("Unknow");
    SnmpLed8.setThresholdValue4(4);
    SnmpLed8.setThresholdValue5(5);
    SnmpLed8.setThresholdValue6(6);
    SnmpLed8.setThresholdValue7(7);
```

```
        SnmpLed8.setThresholdLabel2("Monomode-Fiber");
        SnmpLed8.setThresholdLabel3("Multimode-Fiber");
        SnmpLed8.setThresholdLabel4("Twisted-Pair");
        SnmpLed8.setThresholdLabel5("UTP");
        SnmpLed8.setThresholdLabel6("STP");
        SnmpLed8.setThresholdLabel7("Coaxial Cable");
        SnmpLed8.setNumberOfStates(7);
        SnmpLed8.setPollInterval(300);
        SnmpLed8.setBgColor(new Color(-1));
        SnmpLed8.setFgColor(new Color(-16777216));
        SnmpLed8.setThresholdColor1(new Color(-1));
        SnmpLed8.setThresholdColor2(new Color(-1));
        SnmpLed8.setThresholdColor3(new Color(-1));
        SnmpLed8.setThresholdColor4(new Color(-1));
        SnmpLed8.setThresholdColor5(new Color(-1));
        SnmpLed8.setThresholdColor6(new Color(-1));
        SnmpLed8.setThresholdColor7(new Color(-1));
        SnmpLed8.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.202");
    }catch(Exception ex){};

    try
    {
        SnmpLed11.setShowValue(false);
        SnmpLed11.setThresholdValue1(1);
        SnmpLed11.setThresholdValue2(2);
        SnmpLed11.setThresholdValue3(3);
        SnmpLed11.setCommunity("painel");
        SnmpLed11.setTargetPort(161);
        SnmpLed11.setTargetHost("150.162.252.20");
        SnmpLed11.setThresholdLabel1("Unknow");
        SnmpLed11.setThresholdValue4(4);
        SnmpLed11.setThresholdValue5(5);
        SnmpLed11.setThresholdValue6(6);
        SnmpLed11.setThresholdValue7(7);
        SnmpLed11.setThresholdLabel2("Private UNI");
        SnmpLed11.setThresholdLabel3("Private NNI");
        SnmpLed11.setThresholdLabel4("Public UNI");
        SnmpLed11.setThresholdLabel5("GSMP");
        SnmpLed11.setThresholdLabel6("Void");
        SnmpLed11.setThresholdLabel7("Auto");
        SnmpLed11.setNumberOfStates(7);
        SnmpLed11.setPollInterval(1);
        SnmpLed11.setBgColor(new Color(-1));
        SnmpLed11.setFgColor(new Color(-16777216));
        SnmpLed11.setThresholdColor1(new Color(-1));
        SnmpLed11.setThresholdColor2(new Color(-1));
        SnmpLed11.setThresholdColor3(new Color(-1));
        SnmpLed11.setThresholdColor4(new Color(-1));
        SnmpLed11.setThresholdColor5(new Color(-1));
        SnmpLed11.setThresholdColor6(new Color(-1));
        SnmpLed11.setThresholdColor7(new Color(-1));
        SnmpLed11.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.203");
    }catch(Exception ex){};

    try
    {
        SnmpLed12.setShowValue(false);
        SnmpLed12.setThresholdValue1(1);
        SnmpLed12.setThresholdValue2(2);
        SnmpLed12.setThresholdValue3(3);
        SnmpLed12.setCommunity("painel");
        SnmpLed12.setTargetPort(161);
        SnmpLed12.setTargetHost("150.162.252.20");
        SnmpLed12.setThresholdLabel1("Unknow");
        SnmpLed12.setThresholdValue4(4);
        SnmpLed12.setThresholdValue5(5);
        SnmpLed12.setThresholdValue6(6);
        SnmpLed12.setThresholdValue7(7);
        SnmpLed12.setThresholdLabel3("Mic");
        SnmpLed12.setThresholdLabel2("Internal");
        SnmpLed12.setThresholdLabel4("SC-Duplex");
        SnmpLed12.setThresholdLabel5("Monomode");
        SnmpLed12.setThresholdLabel6("DB-9");
        SnmpLed12.setThresholdLabel7("RJ-45");
        SnmpLed12.setNumberOfStates(7);
        SnmpLed12.setPollInterval(300);
        SnmpLed12.setBgColor(new Color(-1));
        SnmpLed12.setFgColor(new Color(-16711680));
```

```
    SnmpLed12.setThresholdColor1(new Color(-1));
    SnmpLed12.setThresholdColor2(new Color(-1));
    SnmpLed12.setThresholdColor3(new Color(-1));
    SnmpLed12.setThresholdColor4(new Color(-1));
    SnmpLed12.setThresholdColor5(new Color(-1));
    SnmpLed12.setThresholdColor6(new Color(-1));
    SnmpLed12.setThresholdColor7(new Color(-1));
    SnmpLed12.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.203");
}catch(Exception ex){};

try
{
    SnmpTextField11.setCommunity("painel");
    SnmpTextField11.setTargetHost("150.162.252.20");
    SnmpTextField11.setPollInterval(300);
    SnmpTextField11.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.203");
}catch(Exception ex){};

try
{
    SnmpTextField13.setCommunity("painel");
    SnmpTextField13.setTargetHost("150.162.252.20");
    SnmpTextField13.setPollInterval(300);
    SnmpTextField13.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.204");
}catch(Exception ex){};

try
{
    SnmpTextField15.setCommunity("painel");
    SnmpTextField15.setTargetHost("150.162.252.20");
    SnmpTextField15.setPollInterval(300);
    SnmpTextField15.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.302");
}catch(Exception ex){};

try
{
    SnmpTextField19.setCommunity("painel");
    SnmpTextField19.setTargetHost("150.162.252.20");
    SnmpTextField19.setPollInterval(300);
    SnmpTextField19.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.303");
}catch(Exception ex){};

try
{
    SnmpTextField22.setCommunity("painel");
    SnmpTextField22.setTargetHost("150.162.252.20");
    SnmpTextField22.setPollInterval(300);
    SnmpTextField22.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.401");
}catch(Exception ex){};

try
{
    SnmpTextField23.setCommunity("painel");
    SnmpTextField23.setTargetHost("150.162.252.20");
    SnmpTextField23.setPollInterval(300);
    SnmpTextField23.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.402");
}catch(Exception ex){};

try
{
    SnmpTextField26.setCommunity("painel");
    SnmpTextField26.setTargetHost("150.162.252.20");
    SnmpTextField26.setPollInterval(300);
    SnmpTextField26.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.403");
}catch(Exception ex){};

try
{
    SnmpTextField16.setCommunity("painel");
    SnmpTextField16.setTargetHost("150.162.252.20");
    SnmpTextField16.setPollInterval(300);
    SnmpTextField16.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.404");
}catch(Exception ex){};

try
{
    SnmpTextField17.setCommunity("painel");
    SnmpTextField17.setTargetHost("150.162.252.20");
```

```
    SnmpTextField17.setPollInterval(300);
    SnmpTextField17.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.1501");
}catch(Exception ex){};

try
{
    SnmpTextField14.setCommunity("painel");
    SnmpTextField14.setTargetHost("150.162.252.20");
    SnmpTextField14.setPollInterval(300);
    SnmpTextField14.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.301");
}catch(Exception ex){};

try
{
    SnmpTextField29.setCommunity("painel");
    SnmpTextField29.setTargetHost("150.162.252.20");
    SnmpTextField29.setPollInterval(300);
    SnmpTextField29.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.302");
}catch(Exception ex){};

try
{
    SnmpTextField28.setCommunity("painel");
    SnmpTextField28.setTargetHost("150.162.252.20");
    SnmpTextField28.setPollInterval(300);
    SnmpTextField28.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.204");
}catch(Exception ex){};

try
{
    SnmpTextField30.setCommunity("painel");
    SnmpTextField30.setTargetHost("150.162.252.20");
    SnmpTextField30.setPollInterval(300);
    SnmpTextField30.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.302");
}catch(Exception ex){};

try
{
    SnmpTextField31.setCommunity("painel");
    SnmpTextField31.setTargetHost("150.162.252.20");
    SnmpTextField31.setPollInterval(300);
    SnmpTextField31.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.303");
}catch(Exception ex){};

try
{
    SnmpTextField32.setCommunity("painel");
    SnmpTextField32.setTargetHost("150.162.252.20");
    SnmpTextField32.setPollInterval(300);
    SnmpTextField32.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.304");
}catch(Exception ex){};

try
{
    SnmpTextField33.setCommunity("painel");
    SnmpTextField33.setTargetHost("150.162.252.20");
    SnmpTextField33.setPollInterval(300);
    SnmpTextField33.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.401");
}catch(Exception ex){};

try
{
    SnmpTextField34.setCommunity("painel");
    SnmpTextField34.setTargetHost("150.162.252.20");
    SnmpTextField34.setPollInterval(300);
    SnmpTextField34.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.402");
}catch(Exception ex){};

try
{
    SnmpTextField35.setCommunity("painel");
    SnmpTextField35.setTargetHost("150.162.252.20");
    SnmpTextField35.setPollInterval(300);
    SnmpTextField35.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.403");
}catch(Exception ex){};

try
{
```

```
    SnmpTextField36.setCommunity("painel");
    SnmpTextField36.setTargetHost("150.162.252.20");
    SnmpTextField36.setPollInterval(300);
    SnmpTextField36.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.404");
}catch(Exception ex){};

try
{
    SnmpTextField38.setCommunity("painel");
    SnmpTextField38.setTargetHost("150.162.252.20");
    SnmpTextField38.setPollInterval(300);
    SnmpTextField38.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.1502");
}catch(Exception ex){};

try
{
    SnmpTextField18.setCommunity("painel");
    SnmpTextField18.setTargetHost("150.162.252.20");
    SnmpTextField18.setPollInterval(300);
    SnmpTextField18.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.1701");
}catch(Exception ex){};

try
{
    SnmpTextField25.setCommunity("painel");
    SnmpTextField25.setTargetHost("150.162.252.20");
    SnmpTextField25.setPollInterval(300);
    SnmpTextField25.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.1601");
}catch(Exception ex){};

try
{
    SnmpTextField24.setCommunity("painel");
    SnmpTextField24.setTargetHost("150.162.252.20");
    SnmpTextField24.setPollInterval(300);
    SnmpTextField24.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.1504");
}catch(Exception ex){};

try
{
    SnmpTextField27.setCommunity("painel");
    SnmpTextField27.setTargetHost("150.162.252.20");
    SnmpTextField27.setPollInterval(300);
    SnmpTextField27.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.1503");
}catch(Exception ex){};

try
{
    SnmpTextField21.setCommunity("painel");
    SnmpTextField21.setTargetHost("150.162.252.20");
    SnmpTextField21.setPollInterval(300);
    SnmpTextField21.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.1502");
}catch(Exception ex){};

try
{
    SnmpTextField40.setCommunity("painel");
    SnmpTextField40.setTargetHost("150.162.252.20");
    SnmpTextField40.setPollInterval(300);
    SnmpTextField40.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.1504");
}catch(Exception ex){};

try
{
    SnmpTextField39.setCommunity("painel");
    SnmpTextField39.setTargetHost("150.162.252.20");
    SnmpTextField39.setPollInterval(300);
    SnmpTextField39.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.1503");
}catch(Exception ex){};

try
{
    SnmpTextField41.setCommunity("painel");
    SnmpTextField41.setTargetHost("150.162.252.20");
    SnmpTextField41.setPollInterval(300);
    SnmpTextField41.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.1601");
}catch(Exception ex){};
```

```
try
{
    SnmpTextField42.setCommunity("painel");
    SnmpTextField42.setTargetHost("150.162.252.20");
    SnmpTextField42.setPollInterval(300);
    SnmpTextField42.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.1701");
}catch(Exception ex){};

try
{
    SnmpTextField43.setCommunity("painel");
    SnmpTextField43.setTargetHost("150.162.252.20");
    SnmpTextField43.setPollInterval(300);
    SnmpTextField43.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.204");
}catch(Exception ex){};

try
{
    SnmpTextField45.setCommunity("painel");
    SnmpTextField45.setTargetHost("150.162.252.20");
    SnmpTextField45.setPollInterval(300);
    SnmpTextField45.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.302");
}catch(Exception ex){};

try
{
    SnmpTextField46.setCommunity("painel");
    SnmpTextField46.setTargetHost("150.162.252.20");
    SnmpTextField46.setPollInterval(300);
    SnmpTextField46.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.303");
}catch(Exception ex){};

try
{
    SnmpTextField47.setCommunity("painel");
    SnmpTextField47.setTargetHost("150.162.252.20");
    SnmpTextField47.setPollInterval(300);
    SnmpTextField47.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.304");
}catch(Exception ex){};

try
{
    SnmpTextField48.setCommunity("painel");
    SnmpTextField48.setTargetHost("150.162.252.20");
    SnmpTextField48.setPollInterval(300);
    SnmpTextField48.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.401");
}catch(Exception ex){};

try
{
    SnmpTextField49.setCommunity("painel");
    SnmpTextField49.setTargetHost("150.162.252.20");
    SnmpTextField49.setPollInterval(300);
    SnmpTextField49.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.402");
}catch(Exception ex){};

try
{
    SnmpTextField52.setCommunity("painel");
    SnmpTextField52.setTargetHost("150.162.252.20");
    SnmpTextField52.setPollInterval(300);
    SnmpTextField52.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.404");
}catch(Exception ex){};

try
{
    SnmpTextField53.setCommunity("painel");
    SnmpTextField53.setTargetHost("150.162.252.20");
    SnmpTextField53.setPollInterval(300);
    SnmpTextField53.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.1501");
}catch(Exception ex){};

try
{
    SnmpTextField54.setCommunity("painel");
    SnmpTextField54.setTargetHost("150.162.252.20");
    SnmpTextField54.setPollInterval(300);
    SnmpTextField54.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.1502");
```

```
}catch(Exception ex){};

try
{
    SnmpTextField55.setCommunity("painel");
    SnmpTextField55.setTargetHost("150.162.252.20");
    SnmpTextField55.setPollInterval(300);
    SnmpTextField55.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.1503");
}catch(Exception ex){};

try
{
    SnmpTextField57.setCommunity("painel");
    SnmpTextField57.setTargetHost("150.162.252.20");
    SnmpTextField57.setPollInterval(300);
    SnmpTextField57.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.1601");
}catch(Exception ex){};

try
{
    SnmpTextField58.setCommunity("painel");
    SnmpTextField58.setTargetHost("150.162.252.20");
    SnmpTextField58.setPollInterval(300);
    SnmpTextField58.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.1701");
}catch(Exception ex){};


try
{
    SnmpLed16.setShowValue(false);
    SnmpLed16.setThresholdValue1(1);
    SnmpLed16.setThresholdValue2(2);
    SnmpLed16.setThresholdValue3(3);
    SnmpLed16.setCommunity("painel");
    SnmpLed16.setTargetPort(161);
    SnmpLed16.setTargetHost("150.162.252.20");
    SnmpLed16.setThresholdLabel1("Unknow");
    SnmpLed16.setThresholdValue4(4);
    SnmpLed16.setThresholdValue5(5);
    SnmpLed16.setThresholdValue6(6);
    SnmpLed16.setThresholdValue7(7);
    SnmpLed16.setThresholdLabel2("Private UNI");
    SnmpLed16.setThresholdLabel3("Private NNI");
    SnmpLed16.setThresholdLabel4("Public UNI");
    SnmpLed16.setThresholdLabel5("GSMP");
    SnmpLed16.setThresholdLabel6("Void");
    SnmpLed16.setThresholdLabel7("Auto");
    SnmpLed16.setNumberOfStates(7);
    SnmpLed16.setPollInterval(1);
    SnmpLed16.setBgColor(new Color(-1));
    SnmpLed16.setFgColor(new Color(-16777216));
    SnmpLed16.setThresholdColor1(new Color(-1));
    SnmpLed16.setThresholdColor2(new Color(-1));
    SnmpLed16.setThresholdColor3(new Color(-1));
    SnmpLed16.setThresholdColor4(new Color(-1));
    SnmpLed16.setThresholdColor5(new Color(-1));
    SnmpLed16.setThresholdColor6(new Color(-1));
    SnmpLed16.setThresholdColor7(new Color(-1));
    SnmpLed16.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.204");
}catch(Exception ex){};


try
{
    SnmpLed17.setShowValue(false);
    SnmpLed17.setThresholdValue1(1);
    SnmpLed17.setThresholdValue2(2);
    SnmpLed17.setThresholdValue3(3);
    SnmpLed17.setCommunity("painel");
    SnmpLed17.setTargetPort(161);
    SnmpLed17.setTargetHost("150.162.252.20");
    SnmpLed17.setThresholdLabel1("Unknow");
    SnmpLed17.setThresholdValue4(4);
    SnmpLed17.setThresholdValue5(5);
    SnmpLed17.setThresholdValue6(6);
    SnmpLed17.setThresholdValue7(7);
    SnmpLed17.setThresholdLabel2("Private UNI");
    SnmpLed17.setThresholdLabel3("Private NNI");
```

```
    SnmpLed17.setThresholdLabel4("Public UNI");
    SnmpLed17.setThresholdLabel5("GSMP");
    SnmpLed17.setThresholdLabel6("Void");
    SnmpLed17.setThresholdLabel7("Auto");
    SnmpLed17.setNumberOfStates(7);
    SnmpLed17.setPollInterval(1);
    SnmpLed17.setBgColor(new Color(-1));
    SnmpLed17.setFgColor(new Color(-16777216));
    SnmpLed17.setThresholdColor1(new Color(-1));
    SnmpLed17.setThresholdColor2(new Color(-1));
    SnmpLed17.setThresholdColor3(new Color(-1));
    SnmpLed17.setThresholdColor4(new Color(-1));
    SnmpLed17.setThresholdColor5(new Color(-1));
    SnmpLed17.setThresholdColor6(new Color(-1));
    SnmpLed17.setThresholdColor7(new Color(-1));
    SnmpLed17.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.301");
}catch(Exception ex){};


try
{
    SnmpLed18.setShowValue(false);
    SnmpLed18.setThresholdValue1(1);
    SnmpLed18.setThresholdValue2(2);
    SnmpLed18.setThresholdValue3(3);
    SnmpLed18.setCommunity("painel");
    SnmpLed18.setTargetPort(161);
    SnmpLed18.setTargetHost("150.162.252.20");
    SnmpLed18.setThresholdLabel1("Unknow");
    SnmpLed18.setThresholdValue4(4);
    SnmpLed18.setThresholdValue5(5);
    SnmpLed18.setThresholdValue6(6);
    SnmpLed18.setThresholdValue7(7);
    SnmpLed18.setThresholdLabel2("Private UNI");
    SnmpLed18.setThresholdLabel3("Private NNI");
    SnmpLed18.setThresholdLabel4("Public UNI");
    SnmpLed18.setThresholdLabel5("GSMP");
    SnmpLed18.setThresholdLabel6("Void");
    SnmpLed18.setThresholdLabel7("Auto");
    SnmpLed18.setNumberOfStates(7);
    SnmpLed18.setPollInterval(1);
    SnmpLed18.setBgColor(new Color(-1));
    SnmpLed18.setFgColor(new Color(-16777216));
    SnmpLed18.setThresholdColor1(new Color(-1));
    SnmpLed18.setThresholdColor2(new Color(-1));
    SnmpLed18.setThresholdColor3(new Color(-1));
    SnmpLed18.setThresholdColor4(new Color(-1));
    SnmpLed18.setThresholdColor5(new Color(-1));
    SnmpLed18.setThresholdColor6(new Color(-1));
    SnmpLed18.setThresholdColor7(new Color(-1));
    SnmpLed18.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.302");
}catch(Exception ex){};

try
{
    SnmpLed19.setShowValue(false);
    SnmpLed19.setThresholdValue1(1);
    SnmpLed19.setThresholdValue2(2);
    SnmpLed19.setThresholdValue3(3);
    SnmpLed19.setCommunity("painel");
    SnmpLed19.setTargetPort(161);
    SnmpLed19.setTargetHost("150.162.252.20");
    SnmpLed19.setThresholdLabel1("Unknow");
    SnmpLed19.setThresholdValue4(4);
    SnmpLed19.setThresholdValue5(5);
    SnmpLed19.setThresholdValue6(6);
    SnmpLed19.setThresholdValue7(7);
    SnmpLed19.setThresholdLabel2("Private UNI");
    SnmpLed19.setThresholdLabel3("Private NNI");
    SnmpLed19.setThresholdLabel4("Public UNI");
    SnmpLed19.setThresholdLabel5("GSMP");
    SnmpLed19.setThresholdLabel6("Void");
    SnmpLed19.setThresholdLabel7("Auto");
    SnmpLed19.setNumberOfStates(7);
    SnmpLed19.setPollInterval(1);
    SnmpLed19.setBgColor(new Color(-1));
    SnmpLed19.setFgColor(new Color(-16777216));
    SnmpLed19.setThresholdColor1(new Color(-1));
```

```
    SnmpLed19.setThresholdColor3(new Color(-1));
    SnmpLed19.setThresholdColor4(new Color(-1));
    SnmpLed19.setThresholdColor5(new Color(-1));
    SnmpLed19.setThresholdColor6(new Color(-1));
    SnmpLed19.setThresholdColor7(new Color(-1));
    SnmpLed19.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.303");
}catch(Exception ex){};

try
{
    SnmpLed20.setShowValue(false);
    SnmpLed20.setThresholdValue1(1);
    SnmpLed20.setThresholdValue2(2);
    SnmpLed20.setThresholdValue3(3);
    SnmpLed20.setCommunity("painel");
    SnmpLed20.setTargetPort(161);
    SnmpLed20.setTargetHost("150.162.252.20");
    SnmpLed20.setThresholdLabel1("Unknow");
    SnmpLed20.setThresholdValue4(4);
    SnmpLed20.setThresholdValue5(5);
    SnmpLed20.setThresholdValue6(6);
    SnmpLed20.setThresholdValue7(7);
    SnmpLed20.setThresholdLabel2("Private UNI");
    SnmpLed20.setThresholdLabel3("Private NNI");
    SnmpLed20.setThresholdLabel4("Public UNI");
    SnmpLed20.setThresholdLabel5("GSMP");
    SnmpLed20.setThresholdLabel6("Void");
    SnmpLed20.setThresholdLabel7("Auto");
    SnmpLed20.setNumberOfStates(7);
    SnmpLed20.setPollInterval(1);
    SnmpLed20.setBgColor(new Color(-1));
    SnmpLed20.setFgColor(new Color(-16777216));
    SnmpLed20.setThresholdColor1(new Color(-1));
    SnmpLed20.setThresholdColor2(new Color(-1));
    SnmpLed20.setThresholdColor3(new Color(-1));
    SnmpLed20.setThresholdColor4(new Color(-1));
    SnmpLed20.setThresholdColor5(new Color(-1));
    SnmpLed20.setThresholdColor6(new Color(-1));
    SnmpLed20.setThresholdColor7(new Color(-1));
    SnmpLed20.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.304");
}catch(Exception ex){};

try
{
    SnmpLed21.setShowValue(false);
    SnmpLed21.setThresholdValue1(1);
    SnmpLed21.setThresholdValue2(2);
    SnmpLed21.setThresholdValue3(3);
    SnmpLed21.setCommunity("painel");
    SnmpLed21.setTargetPort(161);
    SnmpLed21.setTargetHost("150.162.252.20");
    SnmpLed21.setThresholdLabel1("Unknow");
    SnmpLed21.setThresholdValue4(4);
    SnmpLed21.setThresholdValue5(5);
    SnmpLed21.setThresholdValue6(6);
    SnmpLed21.setThresholdValue7(7);
    SnmpLed21.setThresholdLabel2("Private UNI");
    SnmpLed21.setThresholdLabel3("Private NNI");
    SnmpLed21.setThresholdLabel4("Public UNI");
    SnmpLed21.setThresholdLabel5("GSMP");
    SnmpLed21.setThresholdLabel6("Void");
    SnmpLed21.setThresholdLabel7("Auto");
    SnmpLed21.setNumberOfStates(7);
    SnmpLed21.setPollInterval(1);
    SnmpLed21.setBgColor(new Color(-1));
    SnmpLed21.setFgColor(new Color(-16777216));
    SnmpLed21.setThresholdColor1(new Color(-1));
    SnmpLed21.setThresholdColor2(new Color(-1));
    SnmpLed21.setThresholdColor3(new Color(-1));
    SnmpLed21.setThresholdColor4(new Color(-1));
    SnmpLed21.setThresholdColor5(new Color(-1));
    SnmpLed21.setThresholdColor6(new Color(-1));
    SnmpLed21.setThresholdColor7(new Color(-1));
    SnmpLed21.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.401");
```

```
{
    SnmpLed22.setShowValue(false);
    SnmpLed22.setThresholdValue1(1);
    SnmpLed22.setThresholdValue2(2);
    SnmpLed22.setThresholdValue3(3);
    SnmpLed22.setCommunity("painel");
    SnmpLed22.setTargetPort(161);
    SnmpLed22.setTargetHost("150.162.252.20");
    SnmpLed22.setThresholdLabel1("Unknow");
    SnmpLed22.setThresholdValue4(4);
    SnmpLed22.setThresholdValue5(5);
    SnmpLed22.setThresholdValue6(6);
    SnmpLed22.setThresholdValue7(7);
    SnmpLed22.setThresholdLabel2("Private UNI");
    SnmpLed22.setThresholdLabel3("Private NNI");
    SnmpLed22.setThresholdLabel4("Public UNI");
    SnmpLed22.setThresholdLabel5("GSMP");
    SnmpLed22.setThresholdLabel6("Void");
    SnmpLed22.setThresholdLabel7("Auto");
    SnmpLed22.setNumberOfStates(7);
    SnmpLed22.setPollInterval(1);
    SnmpLed22.setBgColor(new Color(-1));
    SnmpLed22.setFgColor(new Color(-16777216));
    SnmpLed22.setThresholdColor1(new Color(-1));
    SnmpLed22.setThresholdColor2(new Color(-1));
    SnmpLed22.setThresholdColor3(new Color(-1));
    SnmpLed22.setThresholdColor4(new Color(-1));
    SnmpLed22.setThresholdColor5(new Color(-1));
    SnmpLed22.setThresholdColor6(new Color(-1));
    SnmpLed22.setThresholdColor7(new Color(-1));
    SnmpLed22.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.402");
}catch(Exception ex){};

try
{
    SnmpLed23.setShowValue(false);
    SnmpLed23.setThresholdValue1(1);
    SnmpLed23.setThresholdValue2(2);
    SnmpLed23.setThresholdValue3(3);
    SnmpLed23.setCommunity("painel");
    SnmpLed23.setTargetPort(161);
    SnmpLed23.setTargetHost("150.162.252.20");
    SnmpLed23.setThresholdLabel1("Unknow");
    SnmpLed23.setThresholdValue4(4);
    SnmpLed23.setThresholdValue5(5);
    SnmpLed23.setThresholdValue6(6);
    SnmpLed23.setThresholdValue7(7);
    SnmpLed23.setThresholdLabel2("Private UNI");
    SnmpLed23.setThresholdLabel3("Private NNI");
    SnmpLed23.setThresholdLabel4("Public UNI");
    SnmpLed23.setThresholdLabel5("GSMP");
    SnmpLed23.setThresholdLabel6("Void");
    SnmpLed23.setThresholdLabel7("Auto");
    SnmpLed23.setNumberOfStates(7);
    SnmpLed23.setPollInterval(1);
    SnmpLed23.setBgColor(new Color(-1));
    SnmpLed23.setFgColor(new Color(-16777216));
    SnmpLed23.setThresholdColor1(new Color(-1));
    SnmpLed23.setThresholdColor2(new Color(-1));
    SnmpLed23.setThresholdColor3(new Color(-1));
    SnmpLed23.setThresholdColor4(new Color(-1));
    SnmpLed23.setThresholdColor5(new Color(-1));
    SnmpLed23.setThresholdColor6(new Color(-1));
    SnmpLed23.setThresholdColor7(new Color(-1));
    SnmpLed23.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.403");
}catch(Exception ex){};

try
{
    SnmpLed25.setShowValue(false);
    SnmpLed25.setThresholdValue1(1);
    SnmpLed25.setThresholdValue2(2);
    SnmpLed25.setThresholdValue3(3);
    SnmpLed25.setCommunity("painel");
    SnmpLed25.setTargetPort(161);
    SnmpLed25.setTargetHost("150.162.252.20");
    SnmpLed25.setThresholdLabel1("Unknow");
    SnmpLed25.setThresholdValue4(4);
```

```
    SnmpLed25.setThresholdValue5(5);
    SnmpLed25.setThresholdValue6(6);
    SnmpLed25.setThresholdValue7(7);
    SnmpLed25.setThresholdLabel2("Private UNI");
    SnmpLed25.setThresholdLabel3("Private NNI");
    SnmpLed25.setThresholdLabel4("Public UNI");
    SnmpLed25.setThresholdLabel5("GSMP");
    SnmpLed25.setThresholdLabel6("Void");
    SnmpLed25.setThresholdLabel7("Auto");
    SnmpLed25.setNumberOfStates(7);
    SnmpLed25.setPollInterval(1);
    SnmpLed25.setBgColor(new Color(-1));
    SnmpLed25.setFgColor(new Color(-16777216));
    SnmpLed25.setThresholdColor1(new Color(-1));
    SnmpLed25.setThresholdColor2(new Color(-1));
    SnmpLed25.setThresholdColor3(new Color(-1));
    SnmpLed25.setThresholdColor4(new Color(-1));
    SnmpLed25.setThresholdColor5(new Color(-1));
    SnmpLed25.setThresholdColor6(new Color(-1));
    SnmpLed25.setThresholdColor7(new Color(-1));
    SnmpLed25.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.1501");
}catch(Exception ex){};

try
{
    SnmpLed27.setShowValue(false);
    SnmpLed27.setThresholdValue1(1);
    SnmpLed27.setThresholdValue2(2);
    SnmpLed27.setThresholdValue3(3);
    SnmpLed27.setCommunity("painel");
    SnmpLed27.setTargetPort(161);
    SnmpLed27.setTargetHost("150.162.252.20");
    SnmpLed27.setThresholdLabel1("Unknow");
    SnmpLed27.setThresholdValue4(4);
    SnmpLed27.setThresholdValue5(5);
    SnmpLed27.setThresholdValue6(6);
    SnmpLed27.setThresholdValue7(7);
    SnmpLed27.setThresholdLabel2("Private UNI");
    SnmpLed27.setThresholdLabel3("Private NNI");
    SnmpLed27.setThresholdLabel4("Public UNI");
    SnmpLed27.setThresholdLabel5("GSMP");
    SnmpLed27.setThresholdLabel6("Void");
    SnmpLed27.setThresholdLabel7("Auto");
    SnmpLed27.setNumberOfStates(7);
    SnmpLed27.setPollInterval(1);
    SnmpLed27.setBgColor(new Color(-1));
    SnmpLed27.setFgColor(new Color(-16777216));
    SnmpLed27.setThresholdColor1(new Color(-1));
    SnmpLed27.setThresholdColor2(new Color(-1));
    SnmpLed27.setThresholdColor3(new Color(-1));
    SnmpLed27.setThresholdColor4(new Color(-1));
    SnmpLed27.setThresholdColor5(new Color(-1));
    SnmpLed27.setThresholdColor6(new Color(-1));
    SnmpLed27.setThresholdColor7(new Color(-1));
    SnmpLed27.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.1503");
}catch(Exception ex){};

try
{
    SnmpLed28.setShowValue(false);
    SnmpLed28.setThresholdValue1(1);
    SnmpLed28.setThresholdValue2(2);
    SnmpLed28.setThresholdValue3(3);
    SnmpLed28.setCommunity("painel");
    SnmpLed28.setTargetPort(161);
    SnmpLed28.setTargetHost("150.162.252.20");
    SnmpLed28.setThresholdLabel1("Unknow");
    SnmpLed28.setThresholdValue4(4);
    SnmpLed28.setThresholdValue5(5);
    SnmpLed28.setThresholdValue6(6);
    SnmpLed28.setThresholdValue7(7);
    SnmpLed28.setThresholdLabel2("Private UNI");
    SnmpLed28.setThresholdLabel3("Private NNI");
    SnmpLed28.setThresholdLabel4("Public UNI");
    SnmpLed28.setThresholdLabel5("GSMP");
    SnmpLed28.setThresholdLabel6("Void");
    SnmpLed28.setThresholdLabel7("Auto");
    SnmpLed28.setNumberOfStates(7);
```

```
    SnmpLed28.setPollInterval(1);
    SnmpLed28.setBgColor(new Color(-1));
    SnmpLed28.setFgColor(new Color(-16777216));
    SnmpLed28.setThresholdColor1(new Color(-1));
    SnmpLed28.setThresholdColor2(new Color(-1));
    SnmpLed28.setThresholdColor3(new Color(-1));
    SnmpLed28.setThresholdColor4(new Color(-1));
    SnmpLed28.setThresholdColor5(new Color(-1));
    SnmpLed28.setThresholdColor6(new Color(-1));
    SnmpLed28.setThresholdColor7(new Color(-1));
    SnmpLed28.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.1504");
}catch(Exception ex){};

try
{
    SnmpLed29.setShowValue(false);
    SnmpLed29.setThresholdValue1(1);
    SnmpLed29.setThresholdValue2(2);
    SnmpLed29.setThresholdValue3(3);
    SnmpLed29.setCommunity("painel");
    SnmpLed29.setTargetPort(161);
    SnmpLed29.setTargetHost("150.162.252.20");
    SnmpLed29.setThresholdLabel1("Unknow");
    SnmpLed29.setThresholdValue4(4);
    SnmpLed29.setThresholdValue5(5);
    SnmpLed29.setThresholdValue6(6);
    SnmpLed29.setThresholdValue7(7);
    SnmpLed29.setThresholdLabel2("Private UNI");
    SnmpLed29.setThresholdLabel3("Private NNI");
    SnmpLed29.setThresholdLabel4("Public UNI");
    SnmpLed29.setThresholdLabel5("GSMP");
    SnmpLed29.setThresholdLabel6("Void");
    SnmpLed29.setThresholdLabel7("Auto");
    SnmpLed29.setNumberOfStates(7);
    SnmpLed29.setPollInterval(1);
    SnmpLed29.setBgColor(new Color(-1));
    SnmpLed29.setFgColor(new Color(-16777216));
    SnmpLed29.setThresholdColor1(new Color(-1));
    SnmpLed29.setThresholdColor2(new Color(-1));
    SnmpLed29.setThresholdColor3(new Color(-1));
    SnmpLed29.setThresholdColor4(new Color(-1));
    SnmpLed29.setThresholdColor5(new Color(-1));
    SnmpLed29.setThresholdColor6(new Color(-1));
    SnmpLed29.setThresholdColor7(new Color(-1));
    SnmpLed29.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.1601");
}catch(Exception ex){};

try
{
    SnmpLed30.setShowValue(false);
    SnmpLed30.setThresholdValue1(1);
    SnmpLed30.setThresholdValue2(2);
    SnmpLed30.setThresholdValue3(3);
    SnmpLed30.setCommunity("painel");
    SnmpLed30.setTargetPort(161);
    SnmpLed30.setTargetHost("150.162.252.20");
    SnmpLed30.setThresholdLabel1("Unknow");
    SnmpLed30.setThresholdValue4(4);
    SnmpLed30.setThresholdValue5(5);
    SnmpLed30.setThresholdValue6(6);
    SnmpLed30.setThresholdValue7(7);
    SnmpLed30.setThresholdLabel2("Private UNI");
    SnmpLed30.setThresholdLabel3("Private NNI");
    SnmpLed30.setThresholdLabel4("Public UNI");
    SnmpLed30.setThresholdLabel5("GSMP");
    SnmpLed30.setThresholdLabel6("Void");
    SnmpLed30.setThresholdLabel7("Auto");
    SnmpLed30.setNumberOfStates(7);
    SnmpLed30.setPollInterval(1);
    SnmpLed30.setBgColor(new Color(-1));
    SnmpLed30.setFgColor(new Color(-16777216));
    SnmpLed30.setThresholdColor1(new Color(-1));
    SnmpLed30.setThresholdColor2(new Color(-1));
    SnmpLed30.setThresholdColor3(new Color(-1));
    SnmpLed30.setThresholdColor4(new Color(-1));
    SnmpLed30.setThresholdColor5(new Color(-1));
    SnmpLed30.setThresholdColor6(new Color(-1));
    SnmpLed30.setThresholdColor7(new Color(-1));
```

```
    SnmpLed30.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.1701");
}catch(Exception ex){};

try
{
    SnmpLed31.setShowValue(false);
    SnmpLed31.setThresholdValue1(1);
    SnmpLed31.setThresholdValue2(2);
    SnmpLed31.setThresholdValue3(3);
    SnmpLed31.setCommunity("painel");
    SnmpLed31.setTargetPort(161);
    SnmpLed31.setTargetHost("150.162.252.20");
    SnmpLed31.setThresholdLabel1("Unknow");
    SnmpLed31.setThresholdValue4(4);
    SnmpLed31.setThresholdValue5(5);
    SnmpLed31.setThresholdValue6(6);
    SnmpLed31.setThresholdValue7(7);
    SnmpLed31.setThresholdLabel3("Mic");
    SnmpLed31.setThresholdLabel2("Internal");
    SnmpLed31.setThresholdLabel4("SC-Duplex");
    SnmpLed31.setThresholdLabel5("Monomode");
    SnmpLed31.setThresholdLabel6("DB-9");
    SnmpLed31.setThresholdLabel7("RJ-45");
    SnmpLed31.setNumberOfStates(7);
    SnmpLed31.setPollInterval(300);
    SnmpLed31.setBgColor(new Color(-1));
    SnmpLed31.setFgColor(new Color(-16711680));
    SnmpLed31.setThresholdColor1(new Color(-1));
    SnmpLed31.setThresholdColor2(new Color(-1));
    SnmpLed31.setThresholdColor3(new Color(-1));
    SnmpLed31.setThresholdColor4(new Color(-1));
    SnmpLed31.setThresholdColor5(new Color(-1));
    SnmpLed31.setThresholdColor6(new Color(-1));
    SnmpLed31.setThresholdColor7(new Color(-1));
    SnmpLed31.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.204");
}catch(Exception ex){};

try
{
    SnmpLed32.setShowValue(false);
    SnmpLed32.setThresholdValue1(1);
    SnmpLed32.setThresholdValue2(2);
    SnmpLed32.setThresholdValue3(3);
    SnmpLed32.setCommunity("painel");
    SnmpLed32.setTargetPort(161);
    SnmpLed32.setTargetHost("150.162.252.20");
    SnmpLed32.setThresholdLabel1("Unknow");
    SnmpLed32.setThresholdValue4(4);
    SnmpLed32.setThresholdValue5(5);
    SnmpLed32.setThresholdValue6(6);
    SnmpLed32.setThresholdValue7(7);
    SnmpLed32.setThresholdLabel3("Mic");
    SnmpLed32.setThresholdLabel2("Internal");
    SnmpLed32.setThresholdLabel4("SC-Duplex");
    SnmpLed32.setThresholdLabel5("Monomode");
    SnmpLed32.setThresholdLabel6("DB-9");
    SnmpLed32.setThresholdLabel7("RJ-45");
    SnmpLed32.setNumberOfStates(7);
    SnmpLed32.setPollInterval(300);
    SnmpLed32.setBgColor(new Color(-1));
    SnmpLed32.setFgColor(new Color(-16711680));
    SnmpLed32.setThresholdColor1(new Color(-1));
    SnmpLed32.setThresholdColor2(new Color(-1));
    SnmpLed32.setThresholdColor3(new Color(-1));
    SnmpLed32.setThresholdColor4(new Color(-1));
    SnmpLed32.setThresholdColor5(new Color(-1));
    SnmpLed32.setThresholdColor6(new Color(-1));
    SnmpLed32.setThresholdColor7(new Color(-1));
    SnmpLed32.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.301");
}catch(Exception ex){};

try
{
    SnmpLed33.setShowValue(false);
    SnmpLed33.setThresholdValue1(1);
    SnmpLed33.setThresholdValue2(2);
    SnmpLed33.setThresholdValue3(3);
    SnmpLed33.setCommunity("painel");
```

```
        SnmpLed33.setTargetPort(161);
        SnmpLed33.setTargetHost("150.162.252.20");
        SnmpLed33.setThresholdLabel1("Unknow");
        SnmpLed33.setThresholdValue4(4);
        SnmpLed33.setThresholdValue5(5);
        SnmpLed33.setThresholdValue6(6);
        SnmpLed33.setThresholdValue7(7);
        SnmpLed33.setThresholdLabel3("Mic");
        SnmpLed33.setThresholdLabel2("Internal");
        SnmpLed33.setThresholdLabel4("SC-Duplex");
        SnmpLed33.setThresholdLabel5("Monomode");
        SnmpLed33.setThresholdLabel6("DB-9");
        SnmpLed33.setThresholdLabel7("RJ-45");
        SnmpLed33.setNumberOfStates(7);
        SnmpLed33.setPollInterval(300);
        SnmpLed33.setBgColor(new Color(-1));
        SnmpLed33.setFgColor(new Color(-16711680));
        SnmpLed33.setThresholdColor1(new Color(-1));
        SnmpLed33.setThresholdColor2(new Color(-1));
        SnmpLed33.setThresholdColor3(new Color(-1));
        SnmpLed33.setThresholdColor4(new Color(-1));
        SnmpLed33.setThresholdColor5(new Color(-1));
        SnmpLed33.setThresholdColor6(new Color(-1));
        SnmpLed33.setThresholdColor7(new Color(-1));
        SnmpLed33.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.302");
    }catch(Exception ex){};

    try
    {
        SnmpLed34.setShowValue(false);
        SnmpLed34.setThresholdValue1(1);
        SnmpLed34.setThresholdValue2(2);
        SnmpLed34.setThresholdValue3(3);
        SnmpLed34.setCommunity("painel");
        SnmpLed34.setTargetPort(161);
        SnmpLed34.setTargetHost("150.162.252.20");
        SnmpLed34.setThresholdLabel1("Unknow");
        SnmpLed34.setThresholdValue4(4);
        SnmpLed34.setThresholdValue5(5);
        SnmpLed34.setThresholdValue6(6);
        SnmpLed34.setThresholdValue7(7);
        SnmpLed34.setThresholdLabel3("Mic");
        SnmpLed34.setThresholdLabel2("Internal");
        SnmpLed34.setThresholdLabel4("SC-Duplex");
        SnmpLed34.setThresholdLabel5("Monomode");
        SnmpLed34.setThresholdLabel6("DB-9");
        SnmpLed34.setThresholdLabel7("RJ-45");
        SnmpLed34.setNumberOfStates(7);
        SnmpLed34.setPollInterval(300);
        SnmpLed34.setBgColor(new Color(-1));
        SnmpLed34.setFgColor(new Color(-16711680));
        SnmpLed34.setThresholdColor1(new Color(-1));
        SnmpLed34.setThresholdColor2(new Color(-1));
        SnmpLed34.setThresholdColor3(new Color(-1));
        SnmpLed34.setThresholdColor4(new Color(-1));
        SnmpLed34.setThresholdColor5(new Color(-1));
        SnmpLed34.setThresholdColor6(new Color(-1));
        SnmpLed34.setThresholdColor7(new Color(-1));
        SnmpLed34.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.303");
    }catch(Exception ex){};

    try
    {
        SnmpLed35.setShowValue(false);
        SnmpLed35.setThresholdValue1(1);
        SnmpLed35.setThresholdValue2(2);
        SnmpLed35.setThresholdValue3(3);
        SnmpLed35.setCommunity("painel");
        SnmpLed35.setTargetPort(161);
        SnmpLed35.setTargetHost("150.162.252.20");
        SnmpLed35.setThresholdLabel1("Unknow");
        SnmpLed35.setThresholdValue4(4);
        SnmpLed35.setThresholdValue5(5);
        SnmpLed35.setThresholdValue6(6);
        SnmpLed35.setThresholdValue7(7);
        SnmpLed35.setThresholdLabel3("Mic");
        SnmpLed35.setThresholdLabel2("Internal");
        SnmpLed35.setThresholdLabel4("SC-Duplex");
```

```
    SnmpLed35.setThresholdLabel5("Monomode");
    SnmpLed35.setThresholdLabel6("DB-9");
    SnmpLed35.setThresholdLabel7("RJ-45");
    SnmpLed35.setNumberOfStates(7);
    SnmpLed35.setPollInterval(300);
    SnmpLed35.setBgColor(new Color(-1));
    SnmpLed35.setFgColor(new Color(-16711680));
    SnmpLed35.setThresholdColor1(new Color(-1));
    SnmpLed35.setThresholdColor2(new Color(-1));
    SnmpLed35.setThresholdColor3(new Color(-1));
    SnmpLed35.setThresholdColor4(new Color(-1));
    SnmpLed35.setThresholdColor5(new Color(-1));
    SnmpLed35.setThresholdColor6(new Color(-1));
    SnmpLed35.setThresholdColor7(new Color(-1));
    SnmpLed35.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.304");
}catch(Exception ex){};

try
{
    SnmpLed37.setShowValue(false);
    SnmpLed37.setThresholdValue1(1);
    SnmpLed37.setThresholdValue2(2);
    SnmpLed37.setThresholdValue3(3);
    SnmpLed37.setCommunity("painel");
    SnmpLed37.setTargetPort(161);
    SnmpLed37.setTargetHost("150.162.252.20");
    SnmpLed37.setThresholdLabel1("Unknow");
    SnmpLed37.setThresholdValue4(4);
    SnmpLed37.setThresholdValue5(5);
    SnmpLed37.setThresholdValue6(6);
    SnmpLed37.setThresholdValue7(7);
    SnmpLed37.setThresholdLabel3("Mic");
    SnmpLed37.setThresholdLabel2("Internal");
    SnmpLed37.setThresholdLabel4("SC-Duplex");
    SnmpLed37.setThresholdLabel5("Monomode");
    SnmpLed37.setThresholdLabel6("DB-9");
    SnmpLed37.setThresholdLabel7("RJ-45");
    SnmpLed37.setNumberOfStates(7);
    SnmpLed37.setPollInterval(300);
    SnmpLed37.setBgColor(new Color(-1));
    SnmpLed37.setFgColor(new Color(-16711680));
    SnmpLed37.setThresholdColor1(new Color(-1));
    SnmpLed37.setThresholdColor2(new Color(-1));
    SnmpLed37.setThresholdColor3(new Color(-1));
    SnmpLed37.setThresholdColor4(new Color(-1));
    SnmpLed37.setThresholdColor5(new Color(-1));
    SnmpLed37.setThresholdColor6(new Color(-1));
    SnmpLed37.setThresholdColor7(new Color(-1));
    SnmpLed37.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.402");
}catch(Exception ex){};

try
{
    SnmpLed38.setShowValue(false);
    SnmpLed38.setThresholdValue1(1);
    SnmpLed38.setThresholdValue2(2);
    SnmpLed38.setThresholdValue3(3);
    SnmpLed38.setCommunity("painel");
    SnmpLed38.setTargetPort(161);
    SnmpLed38.setTargetHost("150.162.252.20");
    SnmpLed38.setThresholdLabel1("Unknow");
    SnmpLed38.setThresholdValue4(4);
    SnmpLed38.setThresholdValue5(5);
    SnmpLed38.setThresholdValue6(6);
    SnmpLed38.setThresholdValue7(7);
    SnmpLed38.setThresholdLabel3("Mic");
    SnmpLed38.setThresholdLabel2("Internal");
    SnmpLed38.setThresholdLabel4("SC-Duplex");
    SnmpLed38.setThresholdLabel5("Monomode");
    SnmpLed38.setThresholdLabel6("DB-9");
    SnmpLed38.setThresholdLabel7("RJ-45");
    SnmpLed38.setNumberOfStates(7);
    SnmpLed38.setPollInterval(300);
    SnmpLed38.setBgColor(new Color(-1));
    SnmpLed38.setFgColor(new Color(-16711680));
    SnmpLed38.setThresholdColor1(new Color(-1));
    SnmpLed38.setThresholdColor2(new Color(-1));
    SnmpLed38.setThresholdColor3(new Color(-1));
```

```
    SnmpLed38.setThresholdColor4(new Color(-1));
    SnmpLed38.setThresholdColor5(new Color(-1));
    SnmpLed38.setThresholdColor6(new Color(-1));
    SnmpLed38.setThresholdColor7(new Color(-1));
    SnmpLed38.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.403");
}catch(Exception ex){};

try
{
    SnmpLed39.setShowValue(false);
    SnmpLed39.setThresholdValue1(1);
    SnmpLed39.setThresholdValue2(2);
    SnmpLed39.setThresholdValue3(3);
    SnmpLed39.setCommunity("painel");
    SnmpLed39.setTargetPort(161);
    SnmpLed39.setTargetHost("150.162.252.20");
    SnmpLed39.setThresholdLabel1("Unknow");
    SnmpLed39.setThresholdValue4(4);
    SnmpLed39.setThresholdValue5(5);
    SnmpLed39.setThresholdValue6(6);
    SnmpLed39.setThresholdValue7(7);
    SnmpLed39.setThresholdLabel3("Mic");
    SnmpLed39.setThresholdLabel2("Internal");
    SnmpLed39.setThresholdLabel4("SC-Duplex");
    SnmpLed39.setThresholdLabel5("Monomode");
    SnmpLed39.setThresholdLabel6("DB-9");
    SnmpLed39.setThresholdLabel7("RJ-45");
    SnmpLed39.setNumberOfStates(7);
    SnmpLed39.setPollInterval(300);
    SnmpLed39.setBgColor(new Color(-1));
    SnmpLed39.setFgColor(new Color(-16711680));
    SnmpLed39.setThresholdColor1(new Color(-1));
    SnmpLed39.setThresholdColor2(new Color(-1));
    SnmpLed39.setThresholdColor3(new Color(-1));
    SnmpLed39.setThresholdColor4(new Color(-1));
    SnmpLed39.setThresholdColor5(new Color(-1));
    SnmpLed39.setThresholdColor6(new Color(-1));
    SnmpLed39.setThresholdColor7(new Color(-1));
    SnmpLed39.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.404");
}catch(Exception ex){};

try
{
    SnmpLed55.setShowValue(false);
    SnmpLed55.setThresholdValue1(1);
    SnmpLed55.setThresholdValue2(2);
    SnmpLed55.setThresholdValue3(3);
    SnmpLed55.setCommunity("painel");
    SnmpLed55.setTargetPort(161);
    SnmpLed55.setTargetHost("150.162.252.20");
    SnmpLed55.setThresholdLabel1("Unknow");
    SnmpLed55.setThresholdValue4(4);
    SnmpLed55.setThresholdValue5(5);
    SnmpLed55.setThresholdValue6(6);
    SnmpLed55.setThresholdValue7(7);
    SnmpLed55.setThresholdLabel2("Monomode-Fiber");
    SnmpLed55.setThresholdLabel3("Multimode-Fiber");
    SnmpLed55.setThresholdLabel4("Twisted-Pair");
    SnmpLed55.setThresholdLabel5("UTP");
    SnmpLed55.setThresholdLabel6("STP");
    SnmpLed55.setThresholdLabel7("Coaxial Cable");
    SnmpLed55.setNumberOfStates(7);
    SnmpLed55.setPollInterval(300);
    SnmpLed55.setBgColor(new Color(-1));
    SnmpLed55.setFgColor(new Color(-16777216));
    SnmpLed55.setThresholdColor1(new Color(-1));
    SnmpLed55.setThresholdColor2(new Color(-1));
    SnmpLed55.setThresholdColor3(new Color(-1));
    SnmpLed55.setThresholdColor4(new Color(-1));
    SnmpLed55.setThresholdColor5(new Color(-1));
    SnmpLed55.setThresholdColor6(new Color(-1));
    SnmpLed55.setThresholdColor7(new Color(-1));
    SnmpLed55.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.1501");
}catch(Exception ex){};

try
{
    SnmpLed40.setShowValue(false);
```

```
    SnmpLed40.setThresholdValue1(1);
    SnmpLed40.setThresholdValue2(2);
    SnmpLed40.setThresholdValue3(3);
    SnmpLed40.setCommunity("painel");
    SnmpLed40.setTargetPort(161);
    SnmpLed40.setTargetHost("150.162.252.20");
    SnmpLed40.setThresholdLabel1("Unknow");
    SnmpLed40.setThresholdValue4(4);
    SnmpLed40.setThresholdValue5(5);
    SnmpLed40.setThresholdValue6(6);
    SnmpLed40.setThresholdValue7(7);
    SnmpLed40.setThresholdLabel3("Mic");
    SnmpLed40.setThresholdLabel2("Internal");
    SnmpLed40.setThresholdLabel4("SC-Duplex");
    SnmpLed40.setThresholdLabel5("Monomode");
    SnmpLed40.setThresholdLabel6("DB-9");
    SnmpLed40.setThresholdLabel7("RJ-45");
    SnmpLed40.setNumberOfStates(7);
    SnmpLed40.setPollInterval(300);
    SnmpLed40.setBgColor(new Color(-1));
    SnmpLed40.setFgColor(new Color(-16711680));
    SnmpLed40.setThresholdColor1(new Color(-1));
    SnmpLed40.setThresholdColor2(new Color(-1));
    SnmpLed40.setThresholdColor3(new Color(-1));
    SnmpLed40.setThresholdColor4(new Color(-1));
    SnmpLed40.setThresholdColor5(new Color(-1));
    SnmpLed40.setThresholdColor6(new Color(-1));
    SnmpLed40.setThresholdColor7(new Color(-1));
    SnmpLed40.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.1501");
}catch(Exception ex){};

try
{
    SnmpLed41.setShowValue(false);
    SnmpLed41.setThresholdValue1(1);
    SnmpLed41.setThresholdValue2(2);
    SnmpLed41.setThresholdValue3(3);
    SnmpLed41.setCommunity("painel");
    SnmpLed41.setTargetPort(161);
    SnmpLed41.setTargetHost("150.162.252.20");
    SnmpLed41.setThresholdLabel1("Unknow");
    SnmpLed41.setThresholdValue4(4);
    SnmpLed41.setThresholdValue5(5);
    SnmpLed41.setThresholdValue6(6);
    SnmpLed41.setThresholdValue7(7);
    SnmpLed41.setThresholdLabel3("Mic");
    SnmpLed41.setThresholdLabel2("Internal");
    SnmpLed41.setThresholdLabel4("SC-Duplex");
    SnmpLed41.setThresholdLabel5("Monomode");
    SnmpLed41.setThresholdLabel6("DB-9");
    SnmpLed41.setThresholdLabel7("RJ-45");
    SnmpLed41.setNumberOfStates(7);
    SnmpLed41.setPollInterval(300);
    SnmpLed41.setBgColor(new Color(-1));
    SnmpLed41.setFgColor(new Color(-16711680));
    SnmpLed41.setThresholdColor1(new Color(-1));
    SnmpLed41.setThresholdColor2(new Color(-1));
    SnmpLed41.setThresholdColor3(new Color(-1));
    SnmpLed41.setThresholdColor4(new Color(-1));
    SnmpLed41.setThresholdColor5(new Color(-1));
    SnmpLed41.setThresholdColor6(new Color(-1));
    SnmpLed41.setThresholdColor7(new Color(-1));
    SnmpLed41.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.1502");
}catch(Exception ex){};

try
{
    SnmpLed42.setShowValue(false);
    SnmpLed42.setThresholdValue1(1);
    SnmpLed42.setThresholdValue2(2);
    SnmpLed42.setThresholdValue3(3);
    SnmpLed42.setCommunity("painel");
    SnmpLed42.setTargetPort(161);
    SnmpLed42.setTargetHost("150.162.252.20");
    SnmpLed42.setThresholdLabel1("Unknow");
    SnmpLed42.setThresholdValue4(4);
    SnmpLed42.setThresholdValue5(5);
    SnmpLed42.setThresholdValue6(6);
```

```
    SnmpLed42.setThresholdValue7(7);
    SnmpLed42.setThresholdLabel3("Mic");
    SnmpLed42.setThresholdLabel2("Internal");
    SnmpLed42.setThresholdLabel4("SC-Duplex");
    SnmpLed42.setThresholdLabel5("Monomode");
    SnmpLed42.setThresholdLabel6("DB-9");
    SnmpLed42.setThresholdLabel7("RJ-45");
    SnmpLed42.setNumberOfStates(7);
    SnmpLed42.setPollInterval(300);
    SnmpLed42.setBgColor(new Color(-1));
    SnmpLed42.setFgColor(new Color(-16711680));
    SnmpLed42.setThresholdColor1(new Color(-1));
    SnmpLed42.setThresholdColor2(new Color(-1));
    SnmpLed42.setThresholdColor3(new Color(-1));
    SnmpLed42.setThresholdColor4(new Color(-1));
    SnmpLed42.setThresholdColor5(new Color(-1));
    SnmpLed42.setThresholdColor6(new Color(-1));
    SnmpLed42.setThresholdColor7(new Color(-1));
    SnmpLed42.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.1503");
}catch(Exception ex){};


try
{
    SnmpLed43.setShowValue(false);
    SnmpLed43.setThresholdValue1(1);
    SnmpLed43.setThresholdValue2(2);
    SnmpLed43.setThresholdValue3(3);
    SnmpLed43.setCommunity("painel");
    SnmpLed43.setTargetPort(161);
    SnmpLed43.setTargetHost("150.162.252.20");
    SnmpLed43.setThresholdLabel1("Unknow");
    SnmpLed43.setThresholdValue4(4);
    SnmpLed43.setThresholdValue5(5);
    SnmpLed43.setThresholdValue6(6);
    SnmpLed43.setThresholdValue7(7);
    SnmpLed43.setThresholdLabel3("Mic");
    SnmpLed43.setThresholdLabel2("Internal");
    SnmpLed43.setThresholdLabel4("SC-Duplex");
    SnmpLed43.setThresholdLabel5("Monomode");
    SnmpLed43.setThresholdLabel6("DB-9");
    SnmpLed43.setThresholdLabel7("RJ-45");
    SnmpLed43.setNumberOfStates(7);
    SnmpLed43.setPollInterval(300);
    SnmpLed43.setBgColor(new Color(-1));
    SnmpLed43.setFgColor(new Color(-16711680));
    SnmpLed43.setThresholdColor1(new Color(-1));
    SnmpLed43.setThresholdColor2(new Color(-1));
    SnmpLed43.setThresholdColor3(new Color(-1));
    SnmpLed43.setThresholdColor4(new Color(-1));
    SnmpLed43.setThresholdColor5(new Color(-1));
    SnmpLed43.setThresholdColor6(new Color(-1));
    SnmpLed43.setThresholdColor7(new Color(-1));
    SnmpLed43.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.1504");
}catch(Exception ex){};

try
{
    SnmpLed44.setShowValue(false);
    SnmpLed44.setThresholdValue1(1);
    SnmpLed44.setThresholdValue2(2);
    SnmpLed44.setThresholdValue3(3);
    SnmpLed44.setCommunity("painel");
    SnmpLed44.setTargetPort(161);
    SnmpLed44.setTargetHost("150.162.252.20");
    SnmpLed44.setThresholdLabel1("Unknow");
    SnmpLed44.setThresholdValue4(4);
    SnmpLed44.setThresholdValue5(5);
    SnmpLed44.setThresholdValue6(6);
    SnmpLed44.setThresholdValue7(7);
    SnmpLed44.setThresholdLabel3("Mic");
    SnmpLed44.setThresholdLabel2("Internal");
    SnmpLed44.setThresholdLabel4("SC-Duplex");
    SnmpLed44.setThresholdLabel5("Monomode");
    SnmpLed44.setThresholdLabel6("DB-9");
    SnmpLed44.setThresholdLabel7("RJ-45");
    SnmpLed44.setNumberOfStates(7);
```

```
    SnmpLed44.setPollInterval(300);
    SnmpLed44.setBgColor(new Color(-1));
    SnmpLed44.setFgColor(new Color(-16711680));
    SnmpLed44.setThresholdColor1(new Color(-1));
    SnmpLed44.setThresholdColor2(new Color(-1));
    SnmpLed44.setThresholdColor3(new Color(-1));
    SnmpLed44.setThresholdColor4(new Color(-1));
    SnmpLed44.setThresholdColor5(new Color(-1));
    SnmpLed44.setThresholdColor6(new Color(-1));
    SnmpLed44.setThresholdColor7(new Color(-1));
    SnmpLed44.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.1601");
}catch(Exception ex){};

try
{
    SnmpLed45.setShowValue(false);
    SnmpLed45.setThresholdValue1(1);
    SnmpLed45.setThresholdValue2(2);
    SnmpLed45.setThresholdValue3(3);
    SnmpLed45.setCommunity("painel");
    SnmpLed45.setTargetPort(161);
    SnmpLed45.setTargetHost("150.162.252.20");
    SnmpLed45.setThresholdLabel1("Unknow");
    SnmpLed45.setThresholdValue4(4);
    SnmpLed45.setThresholdValue5(5);
    SnmpLed45.setThresholdValue6(6);
    SnmpLed45.setThresholdValue7(7);
    SnmpLed45.setThresholdLabel3("Mic");
    SnmpLed45.setThresholdLabel2("Internal");
    SnmpLed45.setThresholdLabel4("SC-Duplex");
    SnmpLed45.setThresholdLabel5("Monomode");
    SnmpLed45.setThresholdLabel6("DB-9");
    SnmpLed45.setThresholdLabel7("RJ-45");
    SnmpLed45.setNumberOfStates(7);
    SnmpLed45.setPollInterval(300);
    SnmpLed45.setBgColor(new Color(-1));
    SnmpLed45.setFgColor(new Color(-16711680));
    SnmpLed45.setThresholdColor1(new Color(-1));
    SnmpLed45.setThresholdColor2(new Color(-1));
    SnmpLed45.setThresholdColor3(new Color(-1));
    SnmpLed45.setThresholdColor4(new Color(-1));
    SnmpLed45.setThresholdColor5(new Color(-1));
    SnmpLed45.setThresholdColor6(new Color(-1));
    SnmpLed45.setThresholdColor7(new Color(-1));
    SnmpLed45.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.1701");
}catch(Exception ex){};

try
{
    SnmpLed46.setShowValue(false);
    SnmpLed46.setThresholdValue1(1);
    SnmpLed46.setThresholdValue2(2);
    SnmpLed46.setThresholdValue3(3);
    SnmpLed46.setCommunity("painel");
    SnmpLed46.setTargetPort(161);
    SnmpLed46.setTargetHost("150.162.252.20");
    SnmpLed46.setThresholdLabel1("Unknow");
    SnmpLed46.setThresholdValue4(4);
    SnmpLed46.setThresholdValue5(5);
    SnmpLed46.setThresholdValue6(6);
    SnmpLed46.setThresholdValue7(7);
    SnmpLed46.setThresholdLabel2("Monomode-Fiber");
    SnmpLed46.setThresholdLabel3("Multimode-Fiber");
    SnmpLed46.setThresholdLabel4("Twisted-Pair");
    SnmpLed46.setThresholdLabel5("UTP");
    SnmpLed46.setThresholdLabel6("STP");
    SnmpLed46.setThresholdLabel7("Coaxial Cable");
    SnmpLed46.setNumberOfStates(7);
    SnmpLed46.setPollInterval(300);
    SnmpLed46.setBgColor(new Color(-1));
    SnmpLed46.setFgColor(new Color(-16777216));
    SnmpLed46.setThresholdColor1(new Color(-1));
    SnmpLed46.setThresholdColor2(new Color(-1));
    SnmpLed46.setThresholdColor3(new Color(-1));
    SnmpLed46.setThresholdColor4(new Color(-1));
    SnmpLed46.setThresholdColor5(new Color(-1));
    SnmpLed46.setThresholdColor6(new Color(-1));
```

```
    SnmpLed46.setThresholdColor7(new Color(-1));
    SnmpLed46.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.204");
}catch(Exception ex){};

try
{
    SnmpLed47.setShowValue(false);
    SnmpLed47.setThresholdValue1(1);
    SnmpLed47.setThresholdValue2(2);
    SnmpLed47.setThresholdValue3(3);
    SnmpLed47.setCommunity("painel");
    SnmpLed47.setTargetPort(161);
    SnmpLed47.setTargetHost("150.162.252.20");
    SnmpLed47.setThresholdLabel1("Unknow");
    SnmpLed47.setThresholdValue4(4);
    SnmpLed47.setThresholdValue5(5);
    SnmpLed47.setThresholdValue6(6);
    SnmpLed47.setThresholdValue7(7);
    SnmpLed47.setThresholdLabel2("Monomode-Fiber");
    SnmpLed47.setThresholdLabel3("Multimode-Fiber");
    SnmpLed47.setThresholdLabel4("Twisted-Pair");
    SnmpLed47.setThresholdLabel5("UTP");
    SnmpLed47.setThresholdLabel6("STP");
    SnmpLed47.setThresholdLabel7("Coaxial Cable");
    SnmpLed47.setNumberOfStates(7);
    SnmpLed47.setPollInterval(300);
    SnmpLed47.setBgColor(new Color(-1));
    SnmpLed47.setFgColor(new Color(-16777216));
    SnmpLed47.setThresholdColor1(new Color(-1));
    SnmpLed47.setThresholdColor2(new Color(-1));
    SnmpLed47.setThresholdColor3(new Color(-1));
    SnmpLed47.setThresholdColor4(new Color(-1));
    SnmpLed47.setThresholdColor5(new Color(-1));
    SnmpLed47.setThresholdColor6(new Color(-1));
    SnmpLed47.setThresholdColor7(new Color(-1));
    SnmpLed47.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.301");
}catch(Exception ex){};

try
{
    SnmpLed48.setShowValue(false);
    SnmpLed48.setThresholdValue1(1);
    SnmpLed48.setThresholdValue2(2);
    SnmpLed48.setThresholdValue3(3);
    SnmpLed48.setCommunity("painel");
    SnmpLed48.setTargetPort(161);
    SnmpLed48.setTargetHost("150.162.252.20");
    SnmpLed48.setThresholdLabel1("Unknow");
    SnmpLed48.setThresholdValue4(4);
    SnmpLed48.setThresholdValue5(5);
    SnmpLed48.setThresholdValue6(6);
    SnmpLed48.setThresholdValue7(7);
    SnmpLed48.setThresholdLabel2("Monomode-Fiber");
    SnmpLed48.setThresholdLabel3("Multimode-Fiber");
    SnmpLed48.setThresholdLabel4("Twisted-Pair");
    SnmpLed48.setThresholdLabel5("UTP");
    SnmpLed48.setThresholdLabel6("STP");
    SnmpLed48.setThresholdLabel7("Coaxial Cable");
    SnmpLed48.setNumberOfStates(7);
    SnmpLed48.setPollInterval(300);
    SnmpLed48.setBgColor(new Color(-1));
    SnmpLed48.setFgColor(new Color(-16777216));
    SnmpLed48.setThresholdColor1(new Color(-1));
    SnmpLed48.setThresholdColor2(new Color(-1));
    SnmpLed48.setThresholdColor3(new Color(-1));
    SnmpLed48.setThresholdColor4(new Color(-1));
    SnmpLed48.setThresholdColor5(new Color(-1));
    SnmpLed48.setThresholdColor6(new Color(-1));
    SnmpLed48.setThresholdColor7(new Color(-1));
    SnmpLed48.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.302");
}catch(Exception ex){};

try
{
    SnmpLed50.setShowValue(false);
    SnmpLed50.setThresholdValue1(1);
    SnmpLed50.setThresholdValue2(2);
    SnmpLed50.setThresholdValue3(3);
```

```
    SnmpLed50.setCommunity("painel");
    SnmpLed50.setTargetPort(161);
    SnmpLed50.setTargetHost("150.162.252.20");
    SnmpLed50.setThresholdLabel1("Unknow");
    SnmpLed50.setThresholdValue4(4);
    SnmpLed50.setThresholdValue5(5);
    SnmpLed50.setThresholdValue6(6);
    SnmpLed50.setThresholdValue7(7);
    SnmpLed50.setThresholdLabel2("Monomode-Fiber");
    SnmpLed50.setThresholdLabel3("Multimode-Fiber");
    SnmpLed50.setThresholdLabel4("Twisted-Pair");
    SnmpLed50.setThresholdLabel5("UTP");
    SnmpLed50.setThresholdLabel6("STP");
    SnmpLed50.setThresholdLabel7("Coaxial Cable");
    SnmpLed50.setNumberOfStates(7);
    SnmpLed50.setPollInterval(300);
    SnmpLed50.setBgColor(new Color(-1));
    SnmpLed50.setFgColor(new Color(-16777216));
    SnmpLed50.setThresholdColor1(new Color(-1));
    SnmpLed50.setThresholdColor2(new Color(-1));
    SnmpLed50.setThresholdColor3(new Color(-1));
    SnmpLed50.setThresholdColor4(new Color(-1));
    SnmpLed50.setThresholdColor5(new Color(-1));
    SnmpLed50.setThresholdColor6(new Color(-1));
    SnmpLed50.setThresholdColor7(new Color(-1));
    SnmpLed50.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.304");
}catch(Exception ex){};

try
{
    SnmpLed51.setShowValue(false);
    SnmpLed51.setThresholdValue1(1);
    SnmpLed51.setThresholdValue2(2);
    SnmpLed51.setThresholdValue3(3);
    SnmpLed51.setCommunity("painel");
    SnmpLed51.setTargetPort(161);
    SnmpLed51.setTargetHost("150.162.252.20");
    SnmpLed51.setThresholdLabel1("Unknow");
    SnmpLed51.setThresholdValue4(4);
    SnmpLed51.setThresholdValue5(5);
    SnmpLed51.setThresholdValue6(6);
    SnmpLed51.setThresholdValue7(7);
    SnmpLed51.setThresholdLabel2("Monomode-Fiber");
    SnmpLed51.setThresholdLabel3("Multimode-Fiber");
    SnmpLed51.setThresholdLabel4("Twisted-Pair");
    SnmpLed51.setThresholdLabel5("UTP");
    SnmpLed51.setThresholdLabel6("STP");
    SnmpLed51.setThresholdLabel7("Coaxial Cable");
    SnmpLed51.setNumberOfStates(7);
    SnmpLed51.setPollInterval(300);
    SnmpLed51.setBgColor(new Color(-1));
    SnmpLed51.setFgColor(new Color(-16777216));
    SnmpLed51.setThresholdColor1(new Color(-1));
    SnmpLed51.setThresholdColor2(new Color(-1));
    SnmpLed51.setThresholdColor3(new Color(-1));
    SnmpLed51.setThresholdColor4(new Color(-1));
    SnmpLed51.setThresholdColor5(new Color(-1));
    SnmpLed51.setThresholdColor6(new Color(-1));
    SnmpLed51.setThresholdColor7(new Color(-1));
    SnmpLed51.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.401");
}catch(Exception ex){};

try
{
    SnmpLed52.setShowValue(false);
    SnmpLed52.setThresholdValue1(1);
    SnmpLed52.setThresholdValue2(2);
    SnmpLed52.setThresholdValue3(3);
    SnmpLed52.setCommunity("painel");
    SnmpLed52.setTargetPort(161);
    SnmpLed52.setTargetHost("150.162.252.20");
    SnmpLed52.setThresholdLabel1("Unknow");
    SnmpLed52.setThresholdValue4(4);
    SnmpLed52.setThresholdValue5(5);
    SnmpLed52.setThresholdValue6(6);
    SnmpLed52.setThresholdValue7(7);
    SnmpLed52.setThresholdLabel2("Monomode-Fiber");
    SnmpLed52.setThresholdLabel3("Multimode-Fiber");
```

```
    SnmpLed52.setThresholdLabel4("Twisted-Pair");
    SnmpLed52.setThresholdLabel5("UTP");
    SnmpLed52.setThresholdLabel6("STP");
    SnmpLed52.setThresholdLabel7("Coaxial Cable");
    SnmpLed52.setNumberOfStates(7);
    SnmpLed52.setPollInterval(300);
    SnmpLed52.setBgColor(new Color(-1));
    SnmpLed52.setFgColor(new Color(-16777216));
    SnmpLed52.setThresholdColor1(new Color(-1));
    SnmpLed52.setThresholdColor2(new Color(-1));
    SnmpLed52.setThresholdColor3(new Color(-1));
    SnmpLed52.setThresholdColor4(new Color(-1));
    SnmpLed52.setThresholdColor5(new Color(-1));
    SnmpLed52.setThresholdColor6(new Color(-1));
    SnmpLed52.setThresholdColor7(new Color(-1));
    SnmpLed52.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.402");
}catch(Exception ex){};

try
{
    SnmpLed53.setShowValue(false);
    SnmpLed53.setThresholdValue1(1);
    SnmpLed53.setThresholdValue2(2);
    SnmpLed53.setThresholdValue3(3);
    SnmpLed53.setCommunity("painel");
    SnmpLed53.setTargetPort(161);
    SnmpLed53.setTargetHost("150.162.252.20");
    SnmpLed53.setThresholdLabel1("Unknow");
    SnmpLed53.setThresholdValue4(4);
    SnmpLed53.setThresholdValue5(5);
    SnmpLed53.setThresholdValue6(6);
    SnmpLed53.setThresholdValue7(7);
    SnmpLed53.setThresholdLabel2("Monomode-Fiber");
    SnmpLed53.setThresholdLabel3("Multimode-Fiber");
    SnmpLed53.setThresholdLabel4("Twisted-Pair");
    SnmpLed53.setThresholdLabel5("UTP");
    SnmpLed53.setThresholdLabel6("STP");
    SnmpLed53.setThresholdLabel7("Coaxial Cable");
    SnmpLed53.setNumberOfStates(7);
    SnmpLed53.setPollInterval(300);
    SnmpLed53.setBgColor(new Color(-1));
    SnmpLed53.setFgColor(new Color(-16777216));
    SnmpLed53.setThresholdColor1(new Color(-1));
    SnmpLed53.setThresholdColor2(new Color(-1));
    SnmpLed53.setThresholdColor3(new Color(-1));
    SnmpLed53.setThresholdColor4(new Color(-1));
    SnmpLed53.setThresholdColor5(new Color(-1));
    SnmpLed53.setThresholdColor6(new Color(-1));
    SnmpLed53.setThresholdColor7(new Color(-1));
    SnmpLed53.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.403");
}catch(Exception ex){};

try
{
    SnmpLed56.setShowValue(false);
    SnmpLed56.setThresholdValue1(1);
    SnmpLed56.setThresholdValue2(2);
    SnmpLed56.setThresholdValue3(3);
    SnmpLed56.setCommunity("painel");
    SnmpLed56.setTargetPort(161);
    SnmpLed56.setTargetHost("150.162.252.20");
    SnmpLed56.setThresholdLabel1("Unknow");
    SnmpLed56.setThresholdValue4(4);
    SnmpLed56.setThresholdValue5(5);
    SnmpLed56.setThresholdValue6(6);
    SnmpLed56.setThresholdValue7(7);
    SnmpLed56.setThresholdLabel2("Monomode-Fiber");
    SnmpLed56.setThresholdLabel3("Multimode-Fiber");
    SnmpLed56.setThresholdLabel4("Twisted-Pair");
    SnmpLed56.setThresholdLabel5("UTP");
    SnmpLed56.setThresholdLabel6("STP");
    SnmpLed56.setThresholdLabel7("Coaxial Cable");
    SnmpLed56.setNumberOfStates(7);
    SnmpLed56.setPollInterval(300);
    SnmpLed56.setBgColor(new Color(-1));
    SnmpLed56.setFgColor(new Color(-16777216));
    SnmpLed56.setThresholdColor1(new Color(-1));
    SnmpLed56.setThresholdColor2(new Color(-1));
```

```
    SnmpLed56.setThresholdColor3(new Color(-1));
    SnmpLed56.setThresholdColor4(new Color(-1));
    SnmpLed56.setThresholdColor5(new Color(-1));
    SnmpLed56.setThresholdColor6(new Color(-1));
    SnmpLed56.setThresholdColor7(new Color(-1));
    SnmpLed56.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.1502");
}catch(Exception ex){};

try
{
    SnmpLed57.setShowValue(false);
    SnmpLed57.setThresholdValue1(1);
    SnmpLed57.setThresholdValue2(2);
    SnmpLed57.setThresholdValue3(3);
    SnmpLed57.setCommunity("painel");
    SnmpLed57.setTargetPort(161);
    SnmpLed57.setTargetHost("150.162.252.20");
    SnmpLed57.setThresholdLabel1("Unknow");
    SnmpLed57.setThresholdValue4(4);
    SnmpLed57.setThresholdValue5(5);
    SnmpLed57.setThresholdValue6(6);
    SnmpLed57.setThresholdValue7(7);
    SnmpLed57.setThresholdLabel2("Monomode-Fiber");
    SnmpLed57.setThresholdLabel3("Multimode-Fiber");
    SnmpLed57.setThresholdLabel4("Twisted-Pair");
    SnmpLed57.setThresholdLabel5("UTP");
    SnmpLed57.setThresholdLabel6("STP");
    SnmpLed57.setThresholdLabel7("Coaxial Cable");
    SnmpLed57.setNumberOfStates(7);
    SnmpLed57.setPollInterval(300);
    SnmpLed57.setBgColor(new Color(-1));
    SnmpLed57.setFgColor(new Color(-16777216));
    SnmpLed57.setThresholdColor1(new Color(-1));
    SnmpLed57.setThresholdColor2(new Color(-1));
    SnmpLed57.setThresholdColor3(new Color(-1));
    SnmpLed57.setThresholdColor4(new Color(-1));
    SnmpLed57.setThresholdColor5(new Color(-1));
    SnmpLed57.setThresholdColor6(new Color(-1));
    SnmpLed57.setThresholdColor7(new Color(-1));
    SnmpLed57.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.1503");
}catch(Exception ex){};

try
{
    SnmpLed58.setShowValue(false);
    SnmpLed58.setThresholdValue1(1);
    SnmpLed58.setThresholdValue2(2);
    SnmpLed58.setThresholdValue3(3);
    SnmpLed58.setCommunity("painel");
    SnmpLed58.setTargetPort(161);
    SnmpLed58.setTargetHost("150.162.252.20");
    SnmpLed58.setThresholdLabel1("Unknow");
    SnmpLed58.setThresholdValue4(4);
    SnmpLed58.setThresholdValue5(5);
    SnmpLed58.setThresholdValue6(6);
    SnmpLed58.setThresholdValue7(7);
    SnmpLed58.setThresholdLabel2("Monomode-Fiber");
    SnmpLed58.setThresholdLabel3("Multimode-Fiber");
    SnmpLed58.setThresholdLabel4("Twisted-Pair");
    SnmpLed58.setThresholdLabel5("UTP");
    SnmpLed58.setThresholdLabel6("STP");
    SnmpLed58.setThresholdLabel7("Coaxial Cable");
    SnmpLed58.setNumberOfStates(7);
    SnmpLed58.setPollInterval(300);
    SnmpLed58.setBgColor(new Color(-1));
    SnmpLed58.setFgColor(new Color(-16777216));
    SnmpLed58.setThresholdColor1(new Color(-1));
    SnmpLed58.setThresholdColor2(new Color(-1));
    SnmpLed58.setThresholdColor3(new Color(-1));
    SnmpLed58.setThresholdColor4(new Color(-1));
    SnmpLed58.setThresholdColor5(new Color(-1));
    SnmpLed58.setThresholdColor6(new Color(-1));
    SnmpLed58.setThresholdColor7(new Color(-1));
    SnmpLed58.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.1504");
}catch(Exception ex){};

try
{
```

```
        SnmpLed60.setShowValue(false);
        SnmpLed60.setThresholdValue1(1);
        SnmpLed60.setThresholdValue2(2);
        SnmpLed60.setThresholdValue3(3);
        SnmpLed60.setCommunity("painel");
        SnmpLed60.setTargetPort(161);
        SnmpLed60.setTargetHost("150.162.252.20");
        SnmpLed60.setThresholdLabel1("Unknow");
        SnmpLed60.setThresholdValue4(4);
        SnmpLed60.setThresholdValue5(5);
        SnmpLed60.setThresholdValue6(6);
        SnmpLed60.setThresholdValue7(7);
        SnmpLed60.setThresholdLabel2("Monomode-Fiber");
        SnmpLed60.setThresholdLabel3("Multimode-Fiber");
        SnmpLed60.setThresholdLabel4("Twisted-Pair");
        SnmpLed60.setThresholdLabel5("UTP");
        SnmpLed60.setThresholdLabel6("STP");
        SnmpLed60.setThresholdLabel7("Coaxial Cable");
        SnmpLed60.setNumberOfStates(7);
        SnmpLed60.setPollInterval(300);
        SnmpLed60.setBgColor(new Color(-1));
        SnmpLed60.setFgColor(new Color(-16777216));
        SnmpLed60.setThresholdColor1(new Color(-1));
        SnmpLed60.setThresholdColor2(new Color(-1));
        SnmpLed60.setThresholdColor3(new Color(-1));
        SnmpLed60.setThresholdColor4(new Color(-1));
        SnmpLed60.setThresholdColor5(new Color(-1));
        SnmpLed60.setThresholdColor6(new Color(-1));
        SnmpLed60.setThresholdColor7(new Color(-1));
        SnmpLed60.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.1701");
    }catch(Exception ex){};

    try
    {
        SnmpTextField2.setCommunity("painel");
        SnmpTextField2.setTargetHost("150.162.252.20");
        SnmpTextField2.setPollInterval(300);
        SnmpTextField2.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.202");
    }catch(Exception ex){};


    try
    {
        SnmpTextField44.setCommunity("painel");
        SnmpTextField44.setTargetHost("150.162.252.20");
        SnmpTextField44.setPollInterval(300);
        SnmpTextField44.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.302");
    }catch(Exception ex){};


    try
    {
        SnmpTextField20.setCommunity("painel");
        SnmpTextField20.setTargetHost("150.162.252.20");
        SnmpTextField20.setPollInterval(300);
        SnmpTextField20.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.1.304");
    }catch(Exception ex){};

    try
    {
        SnmpTextField51.setCommunity("painel");
        SnmpTextField51.setTargetHost("150.162.252.20");
        SnmpTextField51.setPollInterval(300);
        SnmpTextField51.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.403");
    }catch(Exception ex){};

    try
    {
        SnmpTextField37.setCommunity("painel");
        SnmpTextField37.setTargetHost("150.162.252.20");
        SnmpTextField37.setPollInterval(300);
        SnmpTextField37.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.2.1501");
    }catch(Exception ex){};

    try
    {
        SnmpTextField56.setCommunity("painel");
        SnmpTextField56.setTargetHost("150.162.252.20");
```

```
    SnmpTextField56.setPollInterval(300);
    SnmpTextField56.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.3.1504");
}catch(Exception ex){};

try
{
    SnmpLed24.setShowValue(false);
    SnmpLed24.setThresholdValue1(1);
    SnmpLed24.setThresholdValue2(2);
    SnmpLed24.setThresholdValue3(3);
    SnmpLed24.setCommunity("painel");
    SnmpLed24.setTargetPort(161);
    SnmpLed24.setTargetHost("150.162.252.20");
    SnmpLed24.setThresholdLabel1("Unknow");
    SnmpLed24.setThresholdValue4(4);
    SnmpLed24.setThresholdValue5(5);
    SnmpLed24.setThresholdValue6(6);
    SnmpLed24.setThresholdValue7(7);
    SnmpLed24.setThresholdLabel2("Private UNI");
    SnmpLed24.setThresholdLabel3("Private NNI");
    SnmpLed24.setThresholdLabel4("Public UNI");
    SnmpLed24.setThresholdLabel5("GSMP");
    SnmpLed24.setThresholdLabel6("Void");
    SnmpLed24.setThresholdLabel7("Auto");
    SnmpLed24.setNumberOfStates(7);
    SnmpLed24.setPollInterval(1);
    SnmpLed24.setBgColor(new Color(-1));
    SnmpLed24.setFgColor(new Color(-16777216));
    SnmpLed24.setThresholdColor1(new Color(-1));
    SnmpLed24.setThresholdColor2(new Color(-1));
    SnmpLed24.setThresholdColor3(new Color(-1));
    SnmpLed24.setThresholdColor4(new Color(-1));
    SnmpLed24.setThresholdColor5(new Color(-1));
    SnmpLed24.setThresholdColor6(new Color(-1));
    SnmpLed24.setThresholdColor7(new Color(-1));
    SnmpLed24.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.404");
}catch(Exception ex){};

try
{
    SnmpLed26.setShowValue(false);
    SnmpLed26.setThresholdValue1(1);
    SnmpLed26.setThresholdValue2(2);
    SnmpLed26.setThresholdValue3(3);
    SnmpLed26.setCommunity("painel");
    SnmpLed26.setTargetPort(161);
    SnmpLed26.setTargetHost("150.162.252.20");
    SnmpLed26.setThresholdLabel1("Unknow");
    SnmpLed26.setThresholdValue4(4);
    SnmpLed26.setThresholdValue5(5);
    SnmpLed26.setThresholdValue6(6);
    SnmpLed26.setThresholdValue7(7);
    SnmpLed26.setThresholdLabel2("Private UNI");
    SnmpLed26.setThresholdLabel3("Private NNI");
    SnmpLed26.setThresholdLabel4("Public UNI");
    SnmpLed26.setThresholdLabel5("GSMP");
    SnmpLed26.setThresholdLabel6("Void");
    SnmpLed26.setThresholdLabel7("Auto");
    SnmpLed26.setNumberOfStates(7);
    SnmpLed26.setPollInterval(1);
    SnmpLed26.setBgColor(new Color(-1));
    SnmpLed26.setFgColor(new Color(-16777216));
    SnmpLed26.setThresholdColor1(new Color(-1));
    SnmpLed26.setThresholdColor2(new Color(-1));
    SnmpLed26.setThresholdColor3(new Color(-1));
    SnmpLed26.setThresholdColor4(new Color(-1));
    SnmpLed26.setThresholdColor5(new Color(-1));
    SnmpLed26.setThresholdColor6(new Color(-1));
    SnmpLed26.setThresholdColor7(new Color(-1));
    SnmpLed26.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.7.1502");
}catch(Exception ex){};

try
{
    SnmpLed36.setShowValue(false);
    SnmpLed36.setThresholdValue1(1);
    SnmpLed36.setThresholdValue2(2);
    SnmpLed36.setThresholdValue3(3);
```

```
    SnmpLed36.setCommunity("painel");
    SnmpLed36.setTargetPort(161);
    SnmpLed36.setTargetHost("150.162.252.20");
    SnmpLed36.setThresholdLabel1("Unknow");
    SnmpLed36.setThresholdValue4(4);
    SnmpLed36.setThresholdValue5(5);
    SnmpLed36.setThresholdValue6(6);
    SnmpLed36.setThresholdValue7(7);
    SnmpLed36.setThresholdLabel3("Mic");
    SnmpLed36.setThresholdLabel2("Internal");
    SnmpLed36.setThresholdLabel4("SC-Duplex");
    SnmpLed36.setThresholdLabel5("Monomode");
    SnmpLed36.setThresholdLabel6("DB-9");
    SnmpLed36.setThresholdLabel7("RJ-45");
    SnmpLed36.setNumberOfStates(7);
    SnmpLed36.setPollInterval(300);
    SnmpLed36.setBgColor(new Color(-1));
    SnmpLed36.setFgColor(new Color(-16711680));
    SnmpLed36.setThresholdColor1(new Color(-1));
    SnmpLed36.setThresholdColor2(new Color(-1));
    SnmpLed36.setThresholdColor3(new Color(-1));
    SnmpLed36.setThresholdColor4(new Color(-1));
    SnmpLed36.setThresholdColor5(new Color(-1));
    SnmpLed36.setThresholdColor6(new Color(-1));
    SnmpLed36.setThresholdColor7(new Color(-1));
    SnmpLed36.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.4.401");
}catch(Exception ex){};

try
{
    SnmpLed49.setShowValue(false);
    SnmpLed49.setThresholdValue1(1);
    SnmpLed49.setThresholdValue2(2);
    SnmpLed49.setThresholdValue3(3);
    SnmpLed49.setCommunity("painel");
    SnmpLed49.setTargetPort(161);
    SnmpLed49.setTargetHost("150.162.252.20");
    SnmpLed49.setThresholdLabel1("Unknow");
    SnmpLed49.setThresholdValue4(4);
    SnmpLed49.setThresholdValue5(5);
    SnmpLed49.setThresholdValue6(6);
    SnmpLed49.setThresholdValue7(7);
    SnmpLed49.setThresholdLabel2("Monomode-Fiber");
    SnmpLed49.setThresholdLabel3("Multimode-Fiber");
    SnmpLed49.setThresholdLabel4("Twisted-Pair");
    SnmpLed49.setThresholdLabel5("UTP");
    SnmpLed49.setThresholdLabel6("STP");
    SnmpLed49.setThresholdLabel7("Coaxial Cable");
    SnmpLed49.setNumberOfStates(7);
    SnmpLed49.setPollInterval(300);
    SnmpLed49.setBgColor(new Color(-1));
    SnmpLed49.setFgColor(new Color(-16777216));
    SnmpLed49.setThresholdColor1(new Color(-1));
    SnmpLed49.setThresholdColor2(new Color(-1));
    SnmpLed49.setThresholdColor3(new Color(-1));
    SnmpLed49.setThresholdColor4(new Color(-1));
    SnmpLed49.setThresholdColor5(new Color(-1));
    SnmpLed49.setThresholdColor6(new Color(-1));
    SnmpLed49.setThresholdColor7(new Color(-1));
    SnmpLed49.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.303");
}catch(Exception ex){};

try
{
    SnmpLed59.setShowValue(false);
    SnmpLed59.setThresholdValue1(1);
    SnmpLed59.setThresholdValue2(2);
    SnmpLed59.setThresholdValue3(3);
    SnmpLed59.setCommunity("painel");
    SnmpLed59.setTargetPort(161);
    SnmpLed59.setTargetHost("150.162.252.20");
    SnmpLed59.setThresholdLabel1("Unknow");
    SnmpLed59.setThresholdValue4(4);
    SnmpLed59.setThresholdValue5(5);
    SnmpLed59.setThresholdValue6(6);
    SnmpLed59.setThresholdValue7(7);
    SnmpLed59.setThresholdLabel2("Monomode-Fiber");
    SnmpLed59.setThresholdLabel3("Multimode-Fiber");
```

```
        SnmpLed59.setThresholdLabel4("Twisted-Pair");
        SnmpLed59.setThresholdLabel5("UTP");
        SnmpLed59.setThresholdLabel6("STP");
        SnmpLed59.setThresholdLabel7("Coaxial Cable");
        SnmpLed59.setNumberOfStates(7);
        SnmpLed59.setPollInterval(300);
        SnmpLed59.setBgColor(new Color(-1));
        SnmpLed59.setFgColor(new Color(-16777216));
        SnmpLed59.setThresholdColor1(new Color(-1));
        SnmpLed59.setThresholdColor2(new Color(-1));
        SnmpLed59.setThresholdColor3(new Color(-1));
        SnmpLed59.setThresholdColor4(new Color(-1));
        SnmpLed59.setThresholdColor5(new Color(-1));
        SnmpLed59.setThresholdColor6(new Color(-1));
        SnmpLed59.setThresholdColor7(new Color(-1));
        SnmpLed59.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.1601");
    }catch(Exception ex){};

    {
        JLabel8.setText("OperState");
    }catch(Exception ex){};

    try
    {
        SnmpLed5.setShowValue(false);
        SnmpLed5.setThresholdValue1(1);
        SnmpLed5.setThresholdValue2(2);
        SnmpLed5.setThresholdValue3(3);
        SnmpLed5.setThresholdColor2(new Color(-65536));
        SnmpLed5.setFgColor(new Color(-1));
        SnmpLed5.setCommunity("painel");
        SnmpLed5.setTargetPort(161);
        SnmpLed5.setTargetHost("150.162.252.20");
        SnmpLed5.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.201");
        SnmpLed5.setThresholdLabel1("Unknow");
        SnmpLed5.setThresholdLabel2("Disable-No Signal");
        SnmpLed5.setThresholdLabel3("Disable-Idle");
        SnmpLed5.setThresholdLabel4("No-Signal");
        SnmpLed5.setThresholdLabel5("Idle");
        SnmpLed5.setThresholdLabel6("In-Service");
        SnmpLed5.setThresholdLabel7("No-Address");
        SnmpLed5.setThresholdValue4(4);
        SnmpLed5.setThresholdValue5(5);
        SnmpLed5.setThresholdValue6(6);
        SnmpLed5.setThresholdValue7(7);
        SnmpLed5.setThresholdColor1(new Color(-4144960));
        SnmpLed5.setThresholdColor3(new Color(-10092391));
        SnmpLed5.setThresholdColor6(new Color(-16724992));
        SnmpLed5.setThresholdColor5(new Color(-154));
        SnmpLed5.setNumberOfStates(7);
        SnmpLed5.setPollInterval(300);
    }catch(Exception ex){};

    try
    {
        SnmpLed10.setShowValue(false);
        SnmpLed10.setThresholdValue1(1);
        SnmpLed10.setThresholdValue2(2);
        SnmpLed10.setThresholdValue3(3);
        SnmpLed10.setThresholdColor2(new Color(-65536));
        SnmpLed10.setFgColor(new Color(-1));
        SnmpLed10.setCommunity("painel");
        SnmpLed10.setTargetPort(161);
        SnmpLed10.setTargetHost("150.162.252.20");
        SnmpLed10.setThresholdLabel1("Unknow");
        SnmpLed10.setThresholdLabel2("Disable-No Signal");
        SnmpLed10.setThresholdLabel3("Disable-Idle");
        SnmpLed10.setThresholdLabel4("No-Signal");
        SnmpLed10.setThresholdLabel5("Idle");
        SnmpLed10.setThresholdLabel6("In-Service");
        SnmpLed10.setThresholdLabel7("No-Address");
        SnmpLed10.setThresholdValue4(4);
        SnmpLed10.setThresholdValue5(5);
        SnmpLed10.setThresholdValue6(6);
        SnmpLed10.setThresholdValue7(7);
        SnmpLed10.setThresholdColor1(new Color(-4144960));
        SnmpLed10.setThresholdColor3(new Color(-10092391));
        SnmpLed10.setThresholdColor6(new Color(-16724992));
```

```
    SnmpLed10.setThresholdColor5(new Color(-154));
    SnmpLed10.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.202");
    SnmpLed10.setNumberOfStates(7);
    SnmpLed10.setPollInterval(300);
}catch(Exception ex){};

try
{
    SnmpLed15.setShowValue(false);
    SnmpLed15.setThresholdValue1(1);
    SnmpLed15.setThresholdValue2(2);
    SnmpLed15.setThresholdValue3(3);
    SnmpLed15.setThresholdColor2(new Color(-65536));
    SnmpLed15.setFgColor(new Color(-1));
    SnmpLed15.setCommunity("painel");
    SnmpLed15.setTargetPort(161);
    SnmpLed15.setTargetHost("150.162.252.20");
    SnmpLed15.setThresholdLabel1("Unknow");
    SnmpLed15.setThresholdLabel2("Disable-No Signal");
    SnmpLed15.setThresholdLabel3("Disable-Idle");
    SnmpLed15.setThresholdLabel4("No-Signal");
    SnmpLed15.setThresholdLabel5("Idle");
    SnmpLed15.setThresholdLabel6("In-Service");
    SnmpLed15.setThresholdLabel7("No-Address");
    SnmpLed15.setThresholdValue4(4);
    SnmpLed15.setThresholdValue5(5);
    SnmpLed15.setThresholdValue6(6);
    SnmpLed15.setThresholdValue7(7);
    SnmpLed15.setThresholdColor1(new Color(-4144960));
    SnmpLed15.setThresholdColor3(new Color(-10092391));
    SnmpLed15.setThresholdColor6(new Color(-16724992));
    SnmpLed15.setThresholdColor5(new Color(-154));
    SnmpLed15.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.203");
    SnmpLed15.setNumberOfStates(7);
    SnmpLed15.setPollInterval(300);
}catch(Exception ex){};

try
{
    SnmpLed76.setShowValue(false);
    SnmpLed76.setThresholdValue1(1);
    SnmpLed76.setThresholdValue2(2);
    SnmpLed76.setThresholdValue3(3);
    SnmpLed76.setThresholdColor2(new Color(-65536));
    SnmpLed76.setFgColor(new Color(-1));
    SnmpLed76.setCommunity("painel");
    SnmpLed76.setTargetPort(161);
    SnmpLed76.setTargetHost("150.162.252.20");
    SnmpLed76.setThresholdLabel1("Unknow");
    SnmpLed76.setThresholdLabel2("Disable-No Signal");
    SnmpLed76.setThresholdLabel3("Disable-Idle");
    SnmpLed76.setThresholdLabel4("No-Signal");
    SnmpLed76.setThresholdLabel5("Idle");
    SnmpLed76.setThresholdLabel6("In-Service");
    SnmpLed76.setThresholdLabel7("No-Address");
    SnmpLed76.setThresholdValue4(4);
    SnmpLed76.setThresholdValue5(5);
    SnmpLed76.setThresholdValue6(6);
    SnmpLed76.setThresholdValue7(7);
    SnmpLed76.setThresholdColor1(new Color(-4144960));
    SnmpLed76.setThresholdColor3(new Color(-10092391));
    SnmpLed76.setThresholdColor6(new Color(-16724992));
    SnmpLed76.setThresholdColor5(new Color(-154));
    SnmpLed76.setNumberOfStates(7);
    SnmpLed76.setPollInterval(300);
    SnmpLed76.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.204");
}catch(Exception ex){};

try
{
    SnmpLed77.setShowValue(false);
    SnmpLed77.setThresholdValue1(1);
    SnmpLed77.setThresholdValue2(2);
    SnmpLed77.setThresholdValue3(3);
    SnmpLed77.setThresholdColor2(new Color(-65536));
    SnmpLed77.setFgColor(new Color(-1));
    SnmpLed77.setCommunity("painel");
    SnmpLed77.setTargetPort(161);
```

```
    SnmpLed77.setTargetHost("150.162.252.20");
    SnmpLed77.setThresholdLabel1("Unknow");
    SnmpLed77.setThresholdLabel2("Disable-No Signal");
    SnmpLed77.setThresholdLabel3("Disable-Idle");
    SnmpLed77.setThresholdLabel4("No-Signal");
    SnmpLed77.setThresholdLabel5("Idle");
    SnmpLed77.setThresholdLabel6("In-Service");
    SnmpLed77.setThresholdLabel7("No-Address");
    SnmpLed77.setThresholdValue4(4);
    SnmpLed77.setThresholdValue5(5);
    SnmpLed77.setThresholdValue6(6);
    SnmpLed77.setThresholdValue7(7);
    SnmpLed77.setThresholdColor1(new Color(-4144960));
    SnmpLed77.setThresholdColor3(new Color(-10092391));
    SnmpLed77.setThresholdColor6(new Color(-16724992));
    SnmpLed77.setThresholdColor5(new Color(-154));
    SnmpLed77.setNumberOfStates(7);
    SnmpLed77.setPollInterval(300);
    SnmpLed77.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.301");
}catch(Exception ex){};

try
{
    SnmpLed78.setShowValue(false);
    SnmpLed78.setThresholdValue1(1);
    SnmpLed78.setThresholdValue2(2);
    SnmpLed78.setThresholdValue3(3);
    SnmpLed78.setThresholdColor2(new Color(-65536));
    SnmpLed78.setFgColor(new Color(-1));
    SnmpLed78.setCommunity("painel");
    SnmpLed78.setTargetPort(161);
    SnmpLed78.setTargetHost("150.162.252.20");
    SnmpLed78.setThresholdLabel1("Unknow");
    SnmpLed78.setThresholdLabel2("Disable-No Signal");
    SnmpLed78.setThresholdLabel3("Disable-Idle");
    SnmpLed78.setThresholdLabel4("No-Signal");
    SnmpLed78.setThresholdLabel5("Idle");
    SnmpLed78.setThresholdLabel6("In-Service");
    SnmpLed78.setThresholdLabel7("No-Address");
    SnmpLed78.setThresholdValue4(4);
    SnmpLed78.setThresholdValue5(5);
    SnmpLed78.setThresholdValue6(6);
    SnmpLed78.setThresholdValue7(7);
    SnmpLed78.setThresholdColor1(new Color(-4144960));
    SnmpLed78.setThresholdColor3(new Color(-10092391));
    SnmpLed78.setThresholdColor6(new Color(-16724992));
    SnmpLed78.setThresholdColor5(new Color(-154));
    SnmpLed78.setNumberOfStates(7);
    SnmpLed78.setPollInterval(300);
    SnmpLed78.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.302");
}catch(Exception ex){};


try
{
    SnmpLed82.setShowValue(false);
    SnmpLed82.setThresholdValue1(1);
    SnmpLed82.setThresholdValue2(2);
    SnmpLed82.setThresholdValue3(3);
    SnmpLed82.setThresholdColor2(new Color(-65536));
    SnmpLed82.setFgColor(new Color(-1));
    SnmpLed82.setCommunity("painel");
    SnmpLed82.setTargetPort(161);
    SnmpLed82.setTargetHost("150.162.252.20");
    SnmpLed82.setThresholdLabel1("Unknow");
    SnmpLed82.setThresholdLabel2("Disable-No Signal");
    SnmpLed82.setThresholdLabel3("Disable-Idle");
    SnmpLed82.setThresholdLabel4("No-Signal");
    SnmpLed82.setThresholdLabel5("Idle");
    SnmpLed82.setThresholdLabel6("In-Service");
    SnmpLed82.setThresholdLabel7("No-Address");
    SnmpLed82.setThresholdValue4(4);
    SnmpLed82.setThresholdValue5(5);
    SnmpLed82.setThresholdValue6(6);
    SnmpLed82.setThresholdValue7(7);
    SnmpLed82.setThresholdColor1(new Color(-4144960));
    SnmpLed82.setThresholdColor3(new Color(-10092391));
    SnmpLed82.setThresholdColor6(new Color(-16724992));
```

```
    SnmpLed82.setThresholdColor5(new Color(-154));
    SnmpLed82.setNumberOfStates(7);
    SnmpLed82.setPollInterval(300);
    SnmpLed82.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.303");
}catch(Exception ex){};

try
{
    SnmpLed80.setShowValue(false);
    SnmpLed80.setThresholdValue1(1);
    SnmpLed80.setThresholdValue2(2);
    SnmpLed80.setThresholdValue3(3);
    SnmpLed80.setThresholdColor2(new Color(-65536));
    SnmpLed80.setFgColor(new Color(-1));
    SnmpLed80.setCommunity("painel");
    SnmpLed80.setTargetPort(161);
    SnmpLed80.setTargetHost("150.162.252.20");
    SnmpLed80.setThresholdLabel1("Unknow");
    SnmpLed80.setThresholdLabel2("Disable-No Signal");
    SnmpLed80.setThresholdLabel3("Disable-Idle");
    SnmpLed80.setThresholdLabel4("No-Signal");
    SnmpLed80.setThresholdLabel5("Idle");
    SnmpLed80.setThresholdLabel6("In-Service");
    SnmpLed80.setThresholdLabel7("No-Address");
    SnmpLed80.setThresholdValue4(4);
    SnmpLed80.setThresholdValue5(5);
    SnmpLed80.setThresholdValue6(6);
    SnmpLed80.setThresholdValue7(7);
    SnmpLed80.setThresholdColor1(new Color(-4144960));
    SnmpLed80.setThresholdColor3(new Color(-10092391));
    SnmpLed80.setThresholdColor6(new Color(-16724992));
    SnmpLed80.setThresholdColor5(new Color(-154));
    SnmpLed80.setNumberOfStates(7);
    SnmpLed80.setPollInterval(300);
    SnmpLed80.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.304");
}catch(Exception ex){};

try
{
    SnmpLed81.setShowValue(false);
    SnmpLed81.setThresholdValue1(1);
    SnmpLed81.setThresholdValue2(2);
    SnmpLed81.setThresholdValue3(3);
    SnmpLed81.setThresholdColor2(new Color(-65536));
    SnmpLed81.setFgColor(new Color(-1));
    SnmpLed81.setCommunity("painel");
    SnmpLed81.setTargetPort(161);
    SnmpLed81.setTargetHost("150.162.252.20");
    SnmpLed81.setThresholdLabel1("Unknow");
    SnmpLed81.setThresholdLabel2("Disable-No Signal");
    SnmpLed81.setThresholdLabel3("Disable-Idle");
    SnmpLed81.setThresholdLabel4("No-Signal");
    SnmpLed81.setThresholdLabel5("Idle");
    SnmpLed81.setThresholdLabel6("In-Service");
    SnmpLed81.setThresholdLabel7("No-Address");
    SnmpLed81.setThresholdValue4(4);
    SnmpLed81.setThresholdValue5(5);
    SnmpLed81.setThresholdValue6(6);
    SnmpLed81.setThresholdValue7(7);
    SnmpLed81.setThresholdColor1(new Color(-4144960));
    SnmpLed81.setThresholdColor3(new Color(-10092391));
    SnmpLed81.setThresholdColor6(new Color(-16724992));
    SnmpLed81.setThresholdColor5(new Color(-154));
    SnmpLed81.setNumberOfStates(7);
    SnmpLed81.setPollInterval(300);
    SnmpLed81.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.401");
}catch(Exception ex){};

try
{
    SnmpLed83.setShowValue(false);
    SnmpLed83.setThresholdValue1(1);
    SnmpLed83.setThresholdValue2(2);
    SnmpLed83.setThresholdValue3(3);
    SnmpLed83.setThresholdColor2(new Color(-65536));
    SnmpLed83.setFgColor(new Color(-1));
    SnmpLed83.setCommunity("painel");
    SnmpLed83.setTargetPort(161);
```

```
    SnmpLed83.setTargetHost("150.162.252.20");
    SnmpLed83.setThresholdLabel1("Unknow");
    SnmpLed83.setThresholdLabel2("Disable-No Signal");
    SnmpLed83.setThresholdLabel3("Disable-Idle");
    SnmpLed83.setThresholdLabel4("No-Signal");
    SnmpLed83.setThresholdLabel5("Idle");
    SnmpLed83.setThresholdLabel6("In-Service");
    SnmpLed83.setThresholdLabel7("No-Address");
    SnmpLed83.setThresholdValue4(4);
    SnmpLed83.setThresholdValue5(5);
    SnmpLed83.setThresholdValue6(6);
    SnmpLed83.setThresholdValue7(7);
    SnmpLed83.setThresholdColor1(new Color(-4144960));
    SnmpLed83.setThresholdColor3(new Color(-10092391));
    SnmpLed83.setThresholdColor6(new Color(-16724992));
    SnmpLed83.setThresholdColor5(new Color(-154));
    SnmpLed83.setNumberOfStates(7);
    SnmpLed83.setPollInterval(300);
    SnmpLed83.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.402");
}catch(Exception ex){};

try
{
    SnmpLed84.setShowValue(false);
    SnmpLed84.setThresholdValue1(1);
    SnmpLed84.setThresholdValue2(2);
    SnmpLed84.setThresholdValue3(3);
    SnmpLed84.setThresholdColor2(new Color(-65536));
    SnmpLed84.setFgColor(new Color(-1));
    SnmpLed84.setCommunity("painel");
    SnmpLed84.setTargetPort(161);
    SnmpLed84.setTargetHost("150.162.252.20");
    SnmpLed84.setThresholdLabel1("Unknow");
    SnmpLed84.setThresholdLabel2("Disable-No Signal");
    SnmpLed84.setThresholdLabel3("Disable-Idle");
    SnmpLed84.setThresholdLabel4("No-Signal");
    SnmpLed84.setThresholdLabel5("Idle");
    SnmpLed84.setThresholdLabel6("In-Service");
    SnmpLed84.setThresholdLabel7("No-Address");
    SnmpLed84.setThresholdValue4(4);
    SnmpLed84.setThresholdValue5(5);
    SnmpLed84.setThresholdValue6(6);
    SnmpLed84.setThresholdValue7(7);
    SnmpLed84.setThresholdColor1(new Color(-4144960));
    SnmpLed84.setThresholdColor3(new Color(-10092391));
    SnmpLed84.setThresholdColor6(new Color(-16724992));
    SnmpLed84.setThresholdColor5(new Color(-154));
    SnmpLed84.setNumberOfStates(7);
    SnmpLed84.setPollInterval(300);
    SnmpLed84.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.403");
}catch(Exception ex){};

try
{
    SnmpLed91.setShowValue(false);
    SnmpLed91.setThresholdValue1(1);
    SnmpLed91.setThresholdValue2(2);
    SnmpLed91.setThresholdValue3(3);
    SnmpLed91.setThresholdColor2(new Color(-65536));
    SnmpLed91.setFgColor(new Color(-1));
    SnmpLed91.setCommunity("painel");
    SnmpLed91.setTargetPort(161);
    SnmpLed91.setTargetHost("150.162.252.20");
    SnmpLed91.setThresholdLabel1("Unknow");
    SnmpLed91.setThresholdLabel2("Disable-No Signal");
    SnmpLed91.setThresholdLabel3("Disable-Idle");
    SnmpLed91.setThresholdLabel4("No-Signal");
    SnmpLed91.setThresholdLabel5("Idle");
    SnmpLed91.setThresholdLabel6("In-Service");
    SnmpLed91.setThresholdLabel7("No-Address");
    SnmpLed91.setThresholdValue4(4);
    SnmpLed91.setThresholdValue5(5);
    SnmpLed91.setThresholdValue6(6);
    SnmpLed91.setThresholdValue7(7);
    SnmpLed91.setThresholdColor1(new Color(-4144960));
    SnmpLed91.setThresholdColor3(new Color(-10092391));
    SnmpLed91.setThresholdColor6(new Color(-16724992));
    SnmpLed91.setThresholdColor5(new Color(-154));
```

```
    SnmpLed91.setNumberOfStates(7);
    SnmpLed91.setPollInterval(300);
    SnmpLed91.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.404");
}catch(Exception ex){};

try
{
    SnmpLed85.setShowValue(false);
    SnmpLed85.setThresholdValue1(1);
    SnmpLed85.setThresholdValue2(2);
    SnmpLed85.setThresholdValue3(3);
    SnmpLed85.setThresholdColor2(new Color(-65536));
    SnmpLed85.setFgColor(new Color(-1));
    SnmpLed85.setCommunity("painel");
    SnmpLed85.setTargetPort(161);
    SnmpLed85.setTargetHost("150.162.252.20");
    SnmpLed85.setThresholdLabel1("Unknow");
    SnmpLed85.setThresholdLabel2("Disable-No Signal");
    SnmpLed85.setThresholdLabel3("Disable-Idle");
    SnmpLed85.setThresholdLabel4("No-Signal");
    SnmpLed85.setThresholdLabel5("Idle");
    SnmpLed85.setThresholdLabel6("In-Service");
    SnmpLed85.setThresholdLabel7("No-Address");
    SnmpLed85.setThresholdValue4(4);
    SnmpLed85.setThresholdValue5(5);
    SnmpLed85.setThresholdValue6(6);
    SnmpLed85.setThresholdValue7(7);
    SnmpLed85.setThresholdColor1(new Color(-4144960));
    SnmpLed85.setThresholdColor3(new Color(-10092391));
    SnmpLed85.setThresholdColor6(new Color(-16724992));
    SnmpLed85.setThresholdColor5(new Color(-154));
    SnmpLed85.setNumberOfStates(7);
    SnmpLed85.setPollInterval(300);
    SnmpLed85.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.1501");
}catch(Exception ex){};

try
{
    SnmpLed86.setShowValue(false);
    SnmpLed86.setThresholdValue1(1);
    SnmpLed86.setThresholdValue2(2);
    SnmpLed86.setThresholdValue3(3);
    SnmpLed86.setThresholdColor2(new Color(-65536));
    SnmpLed86.setFgColor(new Color(-1));
    SnmpLed86.setCommunity("painel");
    SnmpLed86.setTargetPort(161);
    SnmpLed86.setTargetHost("150.162.252.20");
    SnmpLed86.setThresholdLabel1("Unknow");
    SnmpLed86.setThresholdLabel2("Disable-No Signal");
    SnmpLed86.setThresholdLabel3("Disable-Idle");
    SnmpLed86.setThresholdLabel4("No-Signal");
    SnmpLed86.setThresholdLabel5("Idle");
    SnmpLed86.setThresholdLabel6("In-Service");
    SnmpLed86.setThresholdLabel7("No-Address");
    SnmpLed86.setThresholdValue4(4);
    SnmpLed86.setThresholdValue5(5);
    SnmpLed86.setThresholdValue6(6);
    SnmpLed86.setThresholdValue7(7);
    SnmpLed86.setThresholdColor1(new Color(-4144960));
    SnmpLed86.setThresholdColor3(new Color(-10092391));
    SnmpLed86.setThresholdColor6(new Color(-16724992));
    SnmpLed86.setThresholdColor5(new Color(-154));
    SnmpLed86.setNumberOfStates(7);
    SnmpLed86.setPollInterval(300);
    SnmpLed86.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.1502");
}catch(Exception ex){};

try
{
    SnmpLed87.setShowValue(false);
    SnmpLed87.setThresholdValue1(1);
    SnmpLed87.setThresholdValue2(2);
    SnmpLed87.setThresholdValue3(3);
    SnmpLed87.setThresholdColor2(new Color(-65536));
    SnmpLed87.setFgColor(new Color(-1));
    SnmpLed87.setCommunity("painel");
    SnmpLed87.setTargetPort(161);
    SnmpLed87.setTargetHost("150.162.252.20");
```

```
    SnmpLed87.setThresholdLabel1("Unknow");
    SnmpLed87.setThresholdLabel2("Disable-No Signal");
    SnmpLed87.setThresholdLabel3("Disable-Idle");
    SnmpLed87.setThresholdLabel4("No-Signal");
    SnmpLed87.setThresholdLabel5("Idle");
    SnmpLed87.setThresholdLabel6("In-Service");
    SnmpLed87.setThresholdLabel7("No-Address");
    SnmpLed87.setThresholdValue4(4);
    SnmpLed87.setThresholdValue5(5);
    SnmpLed87.setThresholdValue6(6);
    SnmpLed87.setThresholdValue7(7);
    SnmpLed87.setThresholdColor1(new Color(-4144960));
    SnmpLed87.setThresholdColor3(new Color(-10092391));
    SnmpLed87.setThresholdColor6(new Color(-16724992));
    SnmpLed87.setThresholdColor5(new Color(-154));
    SnmpLed87.setNumberOfStates(7);
    SnmpLed87.setPollInterval(300);
    SnmpLed87.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.1503");
}catch(Exception ex){};

try
{
    SnmpLed88.setShowValue(false);
    SnmpLed88.setThresholdValue1(1);
    SnmpLed88.setThresholdValue2(2);
    SnmpLed88.setThresholdValue3(3);
    SnmpLed88.setThresholdColor2(new Color(-65536));
    SnmpLed88.setFgColor(new Color(-1));
    SnmpLed88.setCommunity("painel");
    SnmpLed88.setTargetPort(161);
    SnmpLed88.setTargetHost("150.162.252.20");
    SnmpLed88.setThresholdLabel1("Unknow");
    SnmpLed88.setThresholdLabel2("Disable-No Signal");
    SnmpLed88.setThresholdLabel3("Disable-Idle");
    SnmpLed88.setThresholdLabel4("No-Signal");
    SnmpLed88.setThresholdLabel5("Idle");
    SnmpLed88.setThresholdLabel6("In-Service");
    SnmpLed88.setThresholdLabel7("No-Address");
    SnmpLed88.setThresholdValue4(4);
    SnmpLed88.setThresholdValue5(5);
    SnmpLed88.setThresholdValue6(6);
    SnmpLed88.setThresholdValue7(7);
    SnmpLed88.setThresholdColor1(new Color(-4144960));
    SnmpLed88.setThresholdColor3(new Color(-10092391));
    SnmpLed88.setThresholdColor6(new Color(-16724992));
    SnmpLed88.setThresholdColor5(new Color(-154));
    SnmpLed88.setNumberOfStates(7);
    SnmpLed88.setPollInterval(300);
    SnmpLed88.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.1504");
}catch(Exception ex){};


try
{
    SnmpLed89.setShowValue(false);
    SnmpLed89.setThresholdValue1(1);
    SnmpLed89.setThresholdValue2(2);
    SnmpLed89.setThresholdValue3(3);
    SnmpLed89.setThresholdColor2(new Color(-65536));
    SnmpLed89.setFgColor(new Color(-1));
    SnmpLed89.setCommunity("painel");
    SnmpLed89.setTargetPort(161);
    SnmpLed89.setTargetHost("150.162.252.20");
    SnmpLed89.setThresholdLabel1("Unknow");
    SnmpLed89.setThresholdLabel2("Disable-No Signal");
    SnmpLed89.setThresholdLabel3("Disable-Idle");
    SnmpLed89.setThresholdLabel4("No-Signal");
    SnmpLed89.setThresholdLabel5("Idle");
    SnmpLed89.setThresholdLabel6("In-Service");
    SnmpLed89.setThresholdLabel7("No-Address");
    SnmpLed89.setThresholdValue4(4);
    SnmpLed89.setThresholdValue5(5);
    SnmpLed89.setThresholdValue6(6);
    SnmpLed89.setThresholdValue7(7);
    SnmpLed89.setThresholdColor1(new Color(-4144960));
    SnmpLed89.setThresholdColor3(new Color(-10092391));
    SnmpLed89.setThresholdColor6(new Color(-16724992));
    SnmpLed89.setThresholdColor5(new Color(-154));
```

```
        SnmpLed89.setNumberOfStates(7);
        SnmpLed89.setPollInterval(300);
        SnmpLed89.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.1601");
    }catch(Exception ex){};

    try
    {
        SnmpLed90.setShowValue(false);
        SnmpLed90.setThresholdValue1(1);
        SnmpLed90.setThresholdValue2(2);
        SnmpLed90.setThresholdValue3(3);
        SnmpLed90.setThresholdColor2(new Color(-65536));
        SnmpLed90.setFgColor(new Color(-1));
        SnmpLed90.setCommunity("painel");
        SnmpLed90.setTargetPort(161);
        SnmpLed90.setTargetHost("150.162.252.20");
        SnmpLed90.setThresholdLabel1("Unknow");
        SnmpLed90.setThresholdLabel2("Disable-No Signal");
        SnmpLed90.setThresholdLabel3("Disable-Idle");
        SnmpLed90.setThresholdLabel4("No-Signal");
        SnmpLed90.setThresholdLabel5("Idle");
        SnmpLed90.setThresholdLabel6("In-Service");
        SnmpLed90.setThresholdLabel7("No-Address");
        SnmpLed90.setThresholdValue4(4);
        SnmpLed90.setThresholdValue5(5);
        SnmpLed90.setThresholdValue6(6);
        SnmpLed90.setThresholdValue7(7);
        SnmpLed90.setThresholdColor1(new Color(-4144960));
        SnmpLed90.setThresholdColor3(new Color(-10092391));
        SnmpLed90.setThresholdColor6(new Color(-16724992));
        SnmpLed90.setThresholdColor5(new Color(-154));
        SnmpLed90.setNumberOfStates(7);
        SnmpLed90.setPollInterval(300);
        SnmpLed90.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.6.1701");
    }catch(Exception ex){};

    try
    {
        JLabel3.setText("Speed");
    }catch(Exception ex){};

    try
    {
        SnmpTextField3.setCommunity("painel");
        SnmpTextField3.setTargetHost("150.162.252.20");
        SnmpTextField3.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.201");
        SnmpTextField3.setPollInterval(300);
    }catch(Exception ex){};

    try
    {
        SnmpTextField12.setCommunity("painel");
        SnmpTextField12.setTargetHost("150.162.252.20");
        SnmpTextField12.setPollInterval(300);
        SnmpTextField12.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.203");
    }catch(Exception ex){};

    try
    {
        SnmpTextField60.setCommunity("painel");
        SnmpTextField60.setTargetHost("150.162.252.20");
        SnmpTextField60.setPollInterval(300);
        SnmpTextField60.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.204");
    }catch(Exception ex){};

    try
    {
        SnmpTextField61.setCommunity("painel");
        SnmpTextField61.setTargetHost("150.162.252.20");
        SnmpTextField61.setPollInterval(300);
        SnmpTextField61.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.301");
    }catch(Exception ex){};

    try
    {
        SnmpTextField64.setCommunity("painel");
        SnmpTextField64.setTargetHost("150.162.252.20");
        SnmpTextField64.setPollInterval(300);
```

```
    SnmpTextField64.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.304");
}catch(Exception ex){};

try
{
    SnmpTextField66.setCommunity("painel");
    SnmpTextField66.setTargetHost("150.162.252.20");
    SnmpTextField66.setPollInterval(300);
    SnmpTextField66.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.401");
}catch(Exception ex){};

try
{
    SnmpTextField67.setCommunity("painel");
    SnmpTextField67.setTargetHost("150.162.252.20");
    SnmpTextField67.setPollInterval(300);
    SnmpTextField67.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.402");
}catch(Exception ex){};

try
{
    SnmpTextField68.setCommunity("painel");
    SnmpTextField68.setTargetHost("150.162.252.20");
    SnmpTextField68.setPollInterval(300);
    SnmpTextField68.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.403");
}catch(Exception ex){};

try
{
    SnmpTextField69.setCommunity("painel");
    SnmpTextField69.setTargetHost("150.162.252.20");
    SnmpTextField69.setPollInterval(300);
    SnmpTextField69.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.404");
}catch(Exception ex){};

try
{
    SnmpTextField71.setCommunity("painel");
    SnmpTextField71.setTargetHost("150.162.252.20");
    SnmpTextField71.setPollInterval(300);
    SnmpTextField71.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.1502");
}catch(Exception ex){};

try
{
    SnmpTextField72.setCommunity("painel");
    SnmpTextField72.setTargetHost("150.162.252.20");
    SnmpTextField72.setPollInterval(300);
    SnmpTextField72.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.1503");
}catch(Exception ex){};


try
{
    SnmpTextField73.setCommunity("painel");
    SnmpTextField73.setTargetHost("150.162.252.20");
    SnmpTextField73.setPollInterval(300);
    SnmpTextField73.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.1504");
}catch(Exception ex){};


try
{
    SnmpTextField65.setCommunity("painel");
    SnmpTextField65.setTargetHost("150.162.252.20");
    SnmpTextField65.setPollInterval(300);
    SnmpTextField65.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.203");
}catch(Exception ex){};

try
{
    SnmpLed54.setShowValue(false);
    SnmpLed54.setThresholdValue1(1);
    SnmpLed54.setThresholdValue2(2);
    SnmpLed54.setThresholdValue3(3);
    SnmpLed54.setCommunity("painel");
    SnmpLed54.setTargetPort(161);
    SnmpLed54.setTargetHost("150.162.252.20");
```

```
    SnmpLed54.setThresholdLabel1("Unknow");
    SnmpLed54.setThresholdValue4(4);
    SnmpLed54.setThresholdValue5(5);
    SnmpLed54.setThresholdValue6(6);
    SnmpLed54.setThresholdValue7(7);
    SnmpLed54.setThresholdLabel2("Monomode-Fiber");
    SnmpLed54.setThresholdLabel3("Multimode-Fiber");
    SnmpLed54.setThresholdLabel4("Twisted-Pair");
    SnmpLed54.setThresholdLabel5("UTP");
    SnmpLed54.setThresholdLabel6("STP");
    SnmpLed54.setThresholdLabel7("Coaxial Cable");
    SnmpLed54.setNumberOfStates(7);
    SnmpLed54.setPollInterval(300);
    SnmpLed54.setBgColor(new Color(-1));
    SnmpLed54.setFgColor(new Color(-16777216));
    SnmpLed54.setThresholdColor1(new Color(-1));
    SnmpLed54.setThresholdColor2(new Color(-1));
    SnmpLed54.setThresholdColor3(new Color(-1));
    SnmpLed54.setThresholdColor4(new Color(-1));
    SnmpLed54.setThresholdColor5(new Color(-1));
    SnmpLed54.setThresholdColor6(new Color(-1));
    SnmpLed54.setThresholdColor7(new Color(-1));
    SnmpLed54.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.404");
}catch(Exception ex){};

try
{
    SnmpLed92.setShowValue(false);
    SnmpLed92.setThresholdValue1(1);
    SnmpLed92.setThresholdValue2(2);
    SnmpLed92.setThresholdValue3(3);
    SnmpLed92.setCommunity("painel");
    SnmpLed92.setTargetPort(161);
    SnmpLed92.setTargetHost("150.162.252.20");
    SnmpLed92.setThresholdLabel1("Unknow");
    SnmpLed92.setThresholdValue4(4);
    SnmpLed92.setThresholdValue5(5);
    SnmpLed92.setThresholdValue6(6);
    SnmpLed92.setThresholdValue7(7);
    SnmpLed92.setThresholdLabel2("Monomode-Fiber");
    SnmpLed92.setThresholdLabel3("Multimode-Fiber");
    SnmpLed92.setThresholdLabel4("Twisted-Pair");
    SnmpLed92.setThresholdLabel5("UTP");
    SnmpLed92.setThresholdLabel6("STP");
    SnmpLed92.setThresholdLabel7("Coaxial Cable");
    SnmpLed92.setNumberOfStates(7);
    SnmpLed92.setPollInterval(300);
    SnmpLed92.setBgColor(new Color(-1));
    SnmpLed92.setFgColor(new Color(-16777216));
    SnmpLed92.setThresholdColor1(new Color(-1));
    SnmpLed92.setThresholdColor2(new Color(-1));
    SnmpLed92.setThresholdColor3(new Color(-1));
    SnmpLed92.setThresholdColor4(new Color(-1));
    SnmpLed92.setThresholdColor5(new Color(-1));
    SnmpLed92.setThresholdColor6(new Color(-1));
    SnmpLed92.setThresholdColor7(new Color(-1));
    SnmpLed92.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.8.1501");
}catch(Exception ex){};

try
{
    SnmpTextField74.setCommunity("painel");
    SnmpTextField74.setTargetHost("150.162.252.20");
    SnmpTextField74.setPollInterval(300);
    SnmpTextField74.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.1601");
}catch(Exception ex){};

try
{
    SnmpTextField76.setCommunity("painel");
    SnmpTextField76.setTargetHost("150.162.252.20");
    SnmpTextField76.setPollInterval(300);
    SnmpTextField76.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.1701");
}catch(Exception ex){};

try
{
    SnmpTextField70.setCommunity("painel");
```

```
        SnmpTextField70.setTargetHost("150.162.252.20");
        SnmpTextField70.setPollInterval(300);
        SnmpTextField70.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.1501");
    }catch(Exception ex){};

    try
    {
        SnmpTextField63.setCommunity("painel");
        SnmpTextField63.setTargetHost("150.162.252.20");
        SnmpTextField63.setPollInterval(300);
        SnmpTextField63.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.303");
    }catch(Exception ex){};

    try
    {
        SnmpTextField62.setCommunity("painel");
        SnmpTextField62.setTargetHost("150.162.252.20");
        SnmpTextField62.setPollInterval(300);
        SnmpTextField62.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.303");
    }catch(Exception ex){};

    try
    {
        SnmpTextField8.setCommunity("painel");
        SnmpTextField8.setTargetHost("150.162.252.20");
        SnmpTextField8.setPollInterval(300);
        SnmpTextField8.setObjectID(".1.3.6.1.4.1.2.6.33.1.3.4.1.9.202");
    }catch(Exception ex){};

}



javax.swing.JLabel  JLabel1 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField1 = null;
javax.swing.JLabel  JLabel4 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField4 = null;
javax.swing.JLabel  JLabel5 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField5 = null;
javax.swing.JLabel  JLabel7 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed2 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed3 = null;
javax.swing.JLabel  JLabel2 = null;
javax.swing.JLabel  JLabel6 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed1 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField6 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField9 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField10 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField7 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed6 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed7 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed13 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed8 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed11 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed12 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField11 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField13 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField15 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField19 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField22 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField23 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField26 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField16 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField17 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField14 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField29 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField28 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField30 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField31 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField32 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField33 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField34 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField35 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField36 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField38 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField18 = null;
```

```
com.adventnet.netmonitor.SnmpTextField  SnmpTextField25 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField24 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField27 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField21 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField40 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField39 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField41 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField42 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField43 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField45 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField46 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField47 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField48 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField49 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField52 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField53 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField54 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField55 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField57 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField58 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed16 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed17 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed18 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed19 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed20 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed21 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed22 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed23 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed25 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed27 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed28 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed29 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed30 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed31 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed32 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed33 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed34 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed35 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed37 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed38 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed39 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed55 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed40 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed41 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed42 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed43 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed44 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed45 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed46 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed47 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed48 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed50 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed51 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed52 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed53 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed56 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed57 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed58 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed60 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField2 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField44 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField20 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField51 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField37 = null;
com.adventnet.netmonitor.SnmpTextField  SnmpTextField56 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed24 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed26 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed36 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed49 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed59 = null;
javax.swing.JLabel  JLabel8 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed5 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed10 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed15 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed76 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed77 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed78 = null;
com.adventnet.netmonitor.SnmpLed  SnmpLed82 = null;
```

```
    com.adventnet.netmonitor.SnmpLed  SnmpLed80 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed81 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed83 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed84 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed91 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed85 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed86 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed87 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed88 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed89 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed90 = null;
    javax.swing.JLabel  JLabel3 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField3 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField12 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField60 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField61 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField64 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField66 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField67 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField68 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField69 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField71 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField72 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField73 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField65 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed54 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed92 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField74 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField76 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField70 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField63 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField62 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField8 = null;
}
```

Anexo E: Código HTML da página IBM8265LinkState.html

```
<html>

<head>
<meta http-equiv="Content-Language" content="pt-br">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>SIGMA/WS - ibm8265</title>
</head>

<body>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="15%" align="center"><b><img src="logo.jpg"></b></td>
    <td width="85%" align="center"><p align="center" style="word-spacing: 0; line-
height: 100%; text-indent: 0; margin-left: 0; margin-right: 0; margin-top: 8; margin-
bottom: 0"><b><font size="4">SIGMA/WS - Ferramenta para Gerência de redes ATM, via WWW,
Java e SNMP</font></b></p></td>
  </tr>
</table>
<p align="center"><img border="0" src="ibm8265v.jpg" width="125" height="172"></p>
<p align="center"><b><font size="5">Informações sobre os Link's</font></b></p>

<p align="center">
 <OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="746" height="576"
align="baseline" codebase="http://java.sun.com/products/plugin/1.1/ jinstall-11-
win32.cab #Version=1,1,0,0">
 <PARAM NAME="code" VALUE="IBM8265LinkState.class">
 <PARAM NAME="codebase" VALUE="../classes">
 <PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
 <PARAM NAME="archive"VALUE="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
 <COMMENT>
 <EMBED type="application/x-java-applet" width="746" height="576"
code="IBM8265LinkState.class" codebase="../classes"
archive="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
 <NOEMBED>
 </COMMENT>
 </NOEMBED>
 </EMBED>
 </OBJECT>
</p>

</body>

</html>
```

Anexo E1: Código JAVA do applet IBM8265LinkState.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class IBM8265LinkState extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(746,576);
        initialized = true;

        SnmpImageLabel1= new com.adventnet.netmonitor.SnmpImageLabel(this);
        SnmpImageLabel1.setLayout(null);
        getContentPane().add(SnmpImageLabel1);
        SnmpImageLabel1.setBounds(5,5,730,560);

        SnmpLed14= new com.adventnet.netmonitor.SnmpLed(this);
        SnmpLed14.setLayout(null);
        SnmpImageLabel1.add(SnmpLed14);
        SnmpLed14.setBounds(150,65,35,35);

        SnmpLed9= new com.adventnet.netmonitor.SnmpLed(this);
        SnmpLed9.setLayout(null);
        SnmpImageLabel1.add(SnmpLed9);
        SnmpLed9.setBounds(310,340,35,35);

        SnmpLed7= new com.adventnet.netmonitor.SnmpLed(this);
        SnmpLed7.setLayout(null);
        SnmpImageLabel1.add(SnmpLed7);
        SnmpLed7.setBounds(220,330,35,35);

        SnmpLed13= new com.adventnet.netmonitor.SnmpLed(this);
        SnmpLed13.setLayout(null);
        SnmpImageLabel1.add(SnmpLed13);
        SnmpLed13.setBounds(445,95,35,35);

        SnmpLed12= new com.adventnet.netmonitor.SnmpLed(this);
        SnmpLed12.setLayout(null);
        SnmpImageLabel1.add(SnmpLed12);
        SnmpLed12.setBounds(475,185,35,35);

        SnmpLed11= new com.adventnet.netmonitor.SnmpLed(this);
        SnmpLed11.setLayout(null);
        SnmpImageLabel1.add(SnmpLed11);
        SnmpLed11.setBounds(465,275,35,35);

        SnmpLed10= new com.adventnet.netmonitor.SnmpLed(this);
        SnmpLed10.setLayout(null);
        SnmpImageLabel1.add(SnmpLed10);
        SnmpLed10.setBounds(410,330,35,35);

        try
        {
            SnmpImageLabel1.setImageName("ambiente.jpg");
        }catch(Exception ex){};

        try
        {
            SnmpLed14.setTargetHost("150.162.252.20");
            SnmpLed14.setCommunity("painel");
            SnmpLed14.setPollInterval(10);
            SnmpLed14.setNumberOfStates(2);
            SnmpLed14.setThresholdValue2(2);
            SnmpLed14.setThresholdValue1(1);
            SnmpLed14.setThresholdLabel1("Up");
            SnmpLed14.setThresholdLabel2("Down");
            SnmpLed14.setThresholdColor1(new Color(-16711936));
            SnmpLed14.setThresholdColor2(new Color(-65536));
```

```
    SnmpLed14.setObjectID(".1.3.6.1.2.1.2.2.1.8.1701");
}catch(Exception ex){};

try
{
    SnmpLed9.setTargetHost("150.162.252.20");
    SnmpLed9.setCommunity("painel");
    SnmpLed9.setNumberOfStates(2);
    SnmpLed9.setThresholdValue2(2);
    SnmpLed9.setThresholdValue1(1);
    SnmpLed9.setThresholdLabel1("Up");
    SnmpLed9.setThresholdLabel2("Down");
    SnmpLed9.setThresholdColor1(new Color(-16711936));
    SnmpLed9.setThresholdColor2(new Color(-65536));
    SnmpLed9.setObjectID(".1.3.6.1.2.1.2.2.1.8.202");
    SnmpLed9.setPollInterval(10);
}catch(Exception ex){};

try
{
    SnmpLed7.setTargetHost("150.162.252.20");
    SnmpLed7.setCommunity("painel");
    SnmpLed7.setPollInterval(10);
    SnmpLed7.setObjectID(".1.3.6.1.2.1.2.2.1.8.201");
    SnmpLed7.setNumberOfStates(2);
    SnmpLed7.setThresholdValue2(2);
    SnmpLed7.setThresholdValue1(1);
    SnmpLed7.setThresholdLabel1("Up");
    SnmpLed7.setThresholdLabel2("Down");
    SnmpLed7.setThresholdColor1(new Color(-16711936));
    SnmpLed7.setThresholdColor2(new Color(-65536));
}catch(Exception ex){};

try
{
    SnmpLed13.setTargetHost("150.162.252.20");
    SnmpLed13.setCommunity("painel");
    SnmpLed13.setPollInterval(10);
    SnmpLed13.setNumberOfStates(2);
    SnmpLed13.setThresholdValue2(2);
    SnmpLed13.setThresholdValue1(1);
    SnmpLed13.setThresholdLabel1("Up");
    SnmpLed13.setThresholdLabel2("Down");
    SnmpLed13.setThresholdColor1(new Color(-16711936));
    SnmpLed13.setThresholdColor2(new Color(-65536));
    SnmpLed13.setObjectID(".1.3.6.1.2.1.2.2.1.8.1501");
}catch(Exception ex){};

try
{
    SnmpLed12.setTargetHost("150.162.252.20");
    SnmpLed12.setCommunity("painel");
    SnmpLed12.setPollInterval(10);
    SnmpLed12.setNumberOfStates(2);
    SnmpLed12.setThresholdValue2(2);
    SnmpLed12.setThresholdValue1(1);
    SnmpLed12.setThresholdLabel1("Up");
    SnmpLed12.setThresholdLabel2("Down");
    SnmpLed12.setThresholdColor1(new Color(-16711936));
    SnmpLed12.setThresholdColor2(new Color(-65536));
    SnmpLed12.setObjectID(".1.3.6.1.2.1.2.2.1.8.1502");
}catch(Exception ex){};

try
{
    SnmpLed11.setTargetHost("150.162.252.20");
    SnmpLed11.setCommunity("painel");
    SnmpLed11.setPollInterval(10);
    SnmpLed11.setNumberOfStates(2);
    SnmpLed11.setThresholdValue2(2);
    SnmpLed11.setThresholdValue1(1);
    SnmpLed11.setThresholdLabel1("Up");
    SnmpLed11.setThresholdLabel2("Down");
    SnmpLed11.setThresholdColor1(new Color(-16711936));
    SnmpLed11.setThresholdColor2(new Color(-65536));
    SnmpLed11.setObjectID(".1.3.6.1.2.1.2.2.1.8.401");
}catch(Exception ex){};
```

```
        try
        {
            SnmpLed10.setTargetHost("150.162.252.20");
            SnmpLed10.setCommunity("painel");
            SnmpLed10.setPollInterval(10);
            SnmpLed10.setNumberOfStates(2);
            SnmpLed10.setThresholdValue2(2);
            SnmpLed10.setThresholdValue1(1);
            SnmpLed10.setThresholdLabel1("Up");
            SnmpLed10.setThresholdLabel2("Down");
            SnmpLed10.setThresholdColor1(new Color(-16711936));
            SnmpLed10.setThresholdColor2(new Color(-65536));
            SnmpLed10.setObjectID(".1.3.6.1.2.1.2.2.1.8.301");
        }catch(Exception ex){};
    }

    com.adventnet.netmonitor.SnmpImageLabel  SnmpImageLabel1 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed14 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed9 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed7 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed13 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed12 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed11 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed10 = null;
}
```

Anexo F: Código HTML da página IBM8265Trafego.html

```
<html>

<head>
<meta http-equiv="Content-Language" content="pt-br">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
1<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>SIGMA/WS - ibm8265</title>
</head>

<body>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="15%" align="center"><b><img src="logo.jpg"></b></td>
    <td width="85%" align="center"><p align="center" style="word-spacing: 0; line-
height: 100%; text-indent: 0; margin-left: 0; margin-right: 0; margin-top: 8; margin-
bottom: 0"><b><font size="4">SIGMA/WS - Ferramenta para Gerência de redes ATM, via WWW,
Java e SNMP</font></b></p></td>
  </tr>
</table>
<p align="center"><img border="0" src="ibm8265v.jpg" width="125" height="172"></p>
<p align="center"><b><font size="5">Informações sobre o Tráfego nos
Link's</font></b></p>

<p align="center">
 <OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="743" height="574"
align="baseline" codebase="http://java.sun.com/products/plugin/1.1/ jinstall-11-
win32.cab #Version=1,1,0,0">
 <PARAM NAME="code" VALUE="IBM8265Trafego.class">
 <PARAM NAME="codebase" VALUE="../classes">
 <PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
 <PARAM NAME="archive"VALUE="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
 <COMMENT>
 <EMBED type="application/x-java-applet" width="743" height="574"
code="IBM8265Trafego.class" codebase="../classes"
archive="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
 <NOEMBED>
 </COMMENT>
 </NOEMBED>
 </EMBED>
 </OBJECT>
</p>

</body>

</html>
```

Anexo F1: Código JAVA do applet IBM8265Trafego.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class IBM8265Trafego extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(743,574);
        initialized = true;

        SnmpImageLabel1= new com.adventnet.netmonitor.SnmpImageLabel(this);
        SnmpImageLabel1.setLayout(null);
        getContentPane().add(SnmpImageLabel1);
        SnmpImageLabel1.setBounds(5,5,730,560);

        SnmpDigDisp8= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp8);
        SnmpDigDisp8.setBounds(410,330,110,30);

        SnmpDigDisp16= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp16);
        SnmpDigDisp16.setBounds(175,295,110,30);

        SnmpDigDisp3= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp3);
        SnmpDigDisp3.setBounds(80,105,110,30);

        SnmpDigDisp11= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp11);
        SnmpDigDisp11.setBounds(120,70,110,30);

        SnmpDigDisp10= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp10);
        SnmpDigDisp10.setBounds(385,95,110,30);

        SnmpDigDisp13= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp13);
        SnmpDigDisp13.setBounds(430,200,110,30);

        SnmpDigDisp2= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp2);
        SnmpDigDisp2.setBounds(395,240,110,30);

        SnmpDigDisp6= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp6);
        SnmpDigDisp6.setBounds(170,330,105,30);

        SnmpDigDisp7= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp7);
        SnmpDigDisp7.setBounds(290,335,110,30);

        SnmpDigDisp14= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp14);
        SnmpDigDisp14.setBounds(410,295,110,30);

        SnmpDigDisp17= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp17);
        SnmpDigDisp17.setBounds(55,305,110,30);

        SnmpDigDisp1= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp1);
        SnmpDigDisp1.setBounds(430,140,110,30);

        SnmpDigDisp15= new com.adventnet.netmonitor.SnmpDigDisp(this);
        SnmpImageLabel1.add(SnmpDigDisp15);
        SnmpDigDisp15.setBounds(290,300,110,30);
```

```
SnmpDigDisp5= new com.adventnet.netmonitor.SnmpDigDisp(this);
SnmpImageLabel1.add(SnmpDigDisp5);
SnmpDigDisp5.setBounds(55,340,110,30);

SnmpPoller1= new com.adventnet.snmp.beans.SnmpPoller(this);

try
{
    SnmpImageLabel1.setImage(true);
    SnmpImageLabel1.setImageName("trafego.jpg");
}catch(Exception ex){};

try
{
    SnmpDigDisp8.setTargetHost("150.162.252.20");
    SnmpDigDisp8.setCommunity("painel");
    SnmpDigDisp8.setObjectID(".1.3.6.1.2.1.2.2.1.16.401");
    SnmpDigDisp8.setPollInterval(300);
}catch(Exception ex){};

try
{
    SnmpDigDisp16.setTargetHost("150.162.252.20");
    SnmpDigDisp16.setCommunity("painel");
    SnmpDigDisp16.setObjectID(".1.3.6.1.2.1.2.2.1.10.202");
    SnmpDigDisp16.setNumDigits(8);
    SnmpDigDisp16.setForeground(new Color(-3407821));
    SnmpDigDisp16.setPollInterval(300);
}catch(Exception ex){};

try
{
    SnmpDigDisp3.setTargetHost("150.162.252.20");
    SnmpDigDisp3.setCommunity("painel");
    SnmpDigDisp3.setObjectID(".1.3.6.1.2.1.2.2.1.16.1701");
    SnmpDigDisp3.setPollInterval(300);
}catch(Exception ex){};

try
{
    SnmpDigDisp11.setTargetHost("150.162.252.20");
    SnmpDigDisp11.setCommunity("painel");
    SnmpDigDisp11.setForeground(new Color(-3407821));
    SnmpDigDisp11.setObjectID(".1.3.6.1.2.1.2.2.1.10.1701");
    SnmpDigDisp11.setPollInterval(300);
}catch(Exception ex){};

try
{
    SnmpDigDisp10.setTargetHost("150.162.252.20");
    SnmpDigDisp10.setCommunity("painel");
    SnmpDigDisp10.setForeground(new Color(-3407821));
    SnmpDigDisp10.setObjectID(".1.3.6.1.2.1.2.2.1.10.1501");
    SnmpDigDisp10.setPollInterval(300);
}catch(Exception ex){};

try
{
    SnmpDigDisp13.setTargetHost("150.162.252.20");
    SnmpDigDisp13.setCommunity("painel");
    SnmpDigDisp13.setForeground(new Color(-3407821));
    SnmpDigDisp13.setObjectID(".1.3.6.1.2.1.2.2.1.10.1502");
    SnmpDigDisp13.setPollInterval(300);
}catch(Exception ex){};

try
{
    SnmpDigDisp2.setCommunity("painel");
    SnmpDigDisp2.setTargetHost("150.162.252.20");
    SnmpDigDisp2.setObjectID(".1.3.6.1.2.1.2.2.1.16.1502");
    SnmpDigDisp2.setPollInterval(300);
}catch(Exception ex){};

try
{
    SnmpDigDisp6.setTargetHost("150.162.252.20");
    SnmpDigDisp6.setCommunity("painel");
    SnmpDigDisp6.setObjectID(".1.3.6.1.2.1.2.2.1.16.202");
```

```
            SnmpDigDisp6.setNumDigits(8);
            SnmpDigDisp6.setPollInterval(300);
        }catch(Exception ex){};

        try
        {
            SnmpDigDisp7.setTargetHost("150.162.252.20");
            SnmpDigDisp7.setCommunity("painel");
            SnmpDigDisp7.setObjectID(".1.3.6.1.2.1.2.2.1.16.301");
            SnmpDigDisp7.setNumDigits(8);
            SnmpDigDisp7.setPollInterval(300);
        }catch(Exception ex){};

        try
        {
            SnmpDigDisp14.setTargetHost("150.162.252.20");
            SnmpDigDisp14.setCommunity("painel");
            SnmpDigDisp14.setForeground(new Color(-3407821));
            SnmpDigDisp14.setObjectID(".1.3.6.1.2.1.2.2.1.10.401");
            SnmpDigDisp14.setPollInterval(300);
        }catch(Exception ex){};

        try
        {
            SnmpDigDisp17.setTargetHost("150.162.252.20");
            SnmpDigDisp17.setCommunity("painel");
            SnmpDigDisp17.setForeground(new Color(-3407821));
            SnmpDigDisp17.setObjectID(".1.3.6.1.2.1.2.2.1.10.201");
            SnmpDigDisp17.setPollInterval(300);
        }catch(Exception ex){};

        try
        {
            SnmpDigDisp1.setTargetHost("150.162.252.20");
            SnmpDigDisp1.setCommunity("painel");
            SnmpDigDisp1.setObjectID(".1.3.6.1.2.1.2.2.1.16.1501");
            SnmpDigDisp1.setPollInterval(300);
        }catch(Exception ex){};

        try
        {
            SnmpDigDisp15.setTargetHost("150.162.252.20");
            SnmpDigDisp15.setCommunity("painel");
            SnmpDigDisp15.setObjectID(".1.3.6.1.2.1.2.2.1.10.301");
            SnmpDigDisp15.setNumDigits(8);
            SnmpDigDisp15.setForeground(new Color(-3407821));
            SnmpDigDisp15.setPollInterval(300);
        }catch(Exception ex){};

        try
        {
            SnmpDigDisp5.setTargetHost("150.162.252.20");
            SnmpDigDisp5.setCommunity("painel");
            SnmpDigDisp5.setObjectID(".1.3.6.1.2.1.2.2.1.16.201");
            SnmpDigDisp5.setPollInterval(300);
        }catch(Exception ex){};

        try
        {
            SnmpPoller1.setTargetHost("150.162.252.20");
            SnmpPoller1.setCommunity("painel");
            SnmpPoller1.setObjectID(".1.3.6.1.2.1.1.3.0");
            java.lang.String[]  objectIDList_array = new java.lang.String[ 1 ];
            objectIDList_array[ 0 ] = "";
            SnmpPoller1.setObjectIDList(objectIDList_array);
            SnmpPoller1.setPollInterval(10);
        }catch(Exception ex){};
    }
    com.adventnet.netmonitor.SnmpImageLabel  SnmpImageLabel1 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp8 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp16 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp3 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp11 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp10 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp13 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp2 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp6 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp7 = null;
```

```
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp14 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp17 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp1 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp15 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp5 = null;
    com.adventnet.snmp.beans.SnmpPoller  SnmpPoller1 = null;
}
```

Anexo G: Código HTML da página IBM8265PacotesIP.html

```
<html>

<head>
<title>SIGMA/WS - ibm8265</title>
</head>

<body>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="15%" align="center"><b><img src="logo.jpg"></b></td>
    <td width="85%" align="center"><p align="center" style="word-spacing: 0; line-
height: 100%; text-indent: 0; margin-left: 0; margin-right: 0; margin-top: 8; margin-
bottom: 0"><b><font size="4">SIGMA/WS - Ferramenta para Gerência de redes ATM, via WWW,
Java e SNMP</font></b></p></td>
  </tr>
</table>
<p align="center"><img border="0" src="ibm8265v.jpg" width="125" height="172"></p>
<p align="center"><b><font size="5">Informações sobre o Tráfego de Pacotes
IP</font></b></p>

<p align="center">
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="776" height="413"
align="baseline" codebase="http://java.sun.com/products/plugin/1.1/ jinstall-11-
win32.cab #Version=1,1,0,0">
<PARAM NAME="code" VALUE="IBM8265PacIP.class">
<PARAM NAME="codebase" VALUE="../classes">
<PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
<PARAM NAME="archive"VALUE="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<COMMENT>
<EMBED type="application/x-java-applet" width="776" height="413"
code="IBM8265PacIP.class" codebase="../classes"
archive="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<NOEMBED>
</COMMENT>
</NOEMBED>
</EMBED>
</OBJECT>

<BR>
<A HREF="http://150.162.60.250:9191/mestrado/html/IBM8265PacotesIPDia.html"
target=_blank width=400 height=400>Monitoração de Pacotes IP (24 Horas)</A>

</p>

</body>

</html>
```

Anexo G1: Código JAVA do applet IBM8265PacIP.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class IBM8265PacIP extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(776,413);
        initialized = true;

        SnmpFilledGraph1= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph1);
        SnmpFilledGraph1.setBounds(10,10,760,190);

        SnmpFilledGraph2= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph2);
        SnmpFilledGraph2.setBounds(10,200,760,200);

        try
        {
            SnmpFilledGraph1.setYGrids(5);
            SnmpFilledGraph1.setAbsoluteTime(true);
            SnmpFilledGraph1.setTargetHost("150.162.252.20");
            SnmpFilledGraph1.setCommunity("painel");
            SnmpFilledGraph1.setXLabel("Tempo");
            SnmpFilledGraph1.setObjectID(".1.3.6.1.2.1.4.3.0");
            SnmpFilledGraph1.setTitle("in Packets");
            SnmpFilledGraph1.setPollInterval(2);
            SnmpFilledGraph1.setXRange(600);
            SnmpFilledGraph1.setXScalePoints(10);
            SnmpFilledGraph1.setXGrids(10);
            SnmpFilledGraph1.setNoOfValues(300);
        }catch(Exception ex){};

        try
        {
            SnmpFilledGraph2.setYGrids(5);
            SnmpFilledGraph2.setAbsoluteTime(true);
            SnmpFilledGraph2.setTargetHost("150.162.252.20");
            SnmpFilledGraph2.setCommunity("painel");
            SnmpFilledGraph2.setXLabel("Tempo");
            SnmpFilledGraph2.setObjectID(".1.3.6.1.2.1.4.10.0");
            SnmpFilledGraph2.setTitle("out Packets");
            SnmpFilledGraph2.setXRange(600);
            SnmpFilledGraph2.setXScalePoints(10);
            SnmpFilledGraph2.setXGrids(10);
            SnmpFilledGraph2.setPollInterval(2);
            SnmpFilledGraph2.setNoOfValues(300);
        }catch(Exception ex){};
    }
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph1 = null;
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph2 = null;
}
```

Anexo H: Código HTML da página IBM8265PacotesIPDia.html

```
<html>

<head>
<title>SIGMA/WS - ibm8265</title>
</head>

<body>

<p align="center">
<B>Monitoração de Pacotes IP (24 Horas)</B>
<OBJECT  classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"  width="785"  height="405"
align="baseline"     codebase="http://java.sun.com/products/plugin/1.1/     jinstall-11-
win32.cab #Version=1,1,0,0">
<PARAM NAME="code" VALUE="IBM8265PacotesIP.class">
<PARAM NAME="codebase" VALUE="../classes">
<PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
<PARAM
NAME="archive"VALUE="BuilderSwing.jar,AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<COMMENT>
<EMBED         type="application/x-java-applet"         width="785"         height="405"
code="IBM8265PacotesIP.class"                                    codebase="../classes"
archive="BuilderSwing.jar,AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<NOEMBED>
</COMMENT>
</NOEMBED>
</EMBED>
</OBJECT>
</p>

</body>

</html>
```

Anexo H1: Código JAVA do applet IBM8265PacotesIP.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class IBM8265PacotesIP extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(785,405);
        initialized = true;

        SnmpFilledGraph1= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph1);
        SnmpFilledGraph1.setBounds(10,5,760,190);

        SnmpFilledGraph2= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph2);
        SnmpFilledGraph2.setBounds(10,195,760,200);

        try
        {
            SnmpFilledGraph1.setXScalePoints(6);
            SnmpFilledGraph1.setXGrids(24);
            SnmpFilledGraph1.setYGrids(5);
            SnmpFilledGraph1.setXRange(86400);
            SnmpFilledGraph1.setAbsoluteTime(true);
            SnmpFilledGraph1.setTargetHost("150.162.252.20");
            SnmpFilledGraph1.setCommunity("painel");
            SnmpFilledGraph1.setXLabel("Tempo");
            SnmpFilledGraph1.setNoOfValues(2880);
            SnmpFilledGraph1.setObjectID(".1.3.6.1.2.1.4.3.0");
            SnmpFilledGraph1.setPollInterval(30);
            SnmpFilledGraph1.setTitle("in Packets");
        }catch(Exception ex){};

        try
        {
            SnmpFilledGraph2.setXScalePoints(6);
            SnmpFilledGraph2.setXGrids(24);
            SnmpFilledGraph2.setYGrids(5);
            SnmpFilledGraph2.setXRange(86400);
            SnmpFilledGraph2.setAbsoluteTime(true);
            SnmpFilledGraph2.setTargetHost("150.162.252.20");
            SnmpFilledGraph2.setCommunity("painel");
            SnmpFilledGraph2.setXLabel("Tempo");
            SnmpFilledGraph2.setNoOfValues(2880);
            SnmpFilledGraph2.setObjectID(".1.3.6.1.2.1.4.10.0");
            SnmpFilledGraph2.setPollInterval(30);
            SnmpFilledGraph2.setTitle("out Packets");
        }catch(Exception ex){};
    }
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph1 = null;
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph2 = null;
}
```

Anexo I: Código HTML da página IBM8265PacotesTCP.html

```
<html>

<head>
<title>SIGMA/WS - ibm8265</title>
</head>

<body>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="15%" align="center"><b><img src="logo.jpg"></b></td>
    <td width="85%" align="center"><p align="center" style="word-spacing: 0; line-
height: 100%; text-indent: 0; margin-left: 0; margin-right: 0; margin-top: 8; margin-
bottom: 0"><b><font size="4">SIGMA/WS - Ferramenta para Gerência de redes ATM, via WWW,
Java e SNMP</font></b></p></td>
  </tr>
</table>
<p align="center"><img border="0" src="ibm8265v.jpg" width="125" height="172"></p>
<p align="center"><b><font size="5">Informações sobre o Tráfego de Segmentos
TCP</font></b></p>

<p align="center">
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="779" height="412"
align="baseline" codebase="http://java.sun.com/products/plugin/1.1/ jinstall-11-
win32.cab #Version=1,1,0,0">
<PARAM NAME="code" VALUE="IBM8265PacTCP.class">
<PARAM NAME="codebase" VALUE="../classes">
<PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
<PARAM NAME="archive"VALUE="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<COMMENT>
<EMBED type="application/x-java-applet" width="779" height="412"
code="IBM8265PacTCP.class" codebase="../classes"
archive="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<NOEMBED>
</COMMENT>
</NOEMBED>
</EMBED>
</OBJECT>

<BR>
<A HREF="http://150.162.60.250:9191/mestrado/html/IBM8265PacotesTCPDia.html"
target=_blank width=400 height=400>Monitoração de Segmentos TCP (24 Horas)</A>

</p>

</body>

</html>
```

Anexo I1: Código JAVA do applet IBM8265PacTCP.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class IBM8265PacTCP extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
          } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(779,412);
        initialized = true;

        SnmpFilledGraph1= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph1);
        SnmpFilledGraph1.setBounds(10,10,760,190);

        SnmpFilledGraph2= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph2);
        SnmpFilledGraph2.setBounds(10,200,760,200);

        try
        {
            SnmpFilledGraph1.setYGrids(5);
            SnmpFilledGraph1.setAbsoluteTime(true);
            SnmpFilledGraph1.setTargetHost("150.162.252.20");
            SnmpFilledGraph1.setCommunity("painel");
            SnmpFilledGraph1.setXLabel("Tempo");
            SnmpFilledGraph1.setTitle("in Packets");
            SnmpFilledGraph1.setPollInterval(2);
            SnmpFilledGraph1.setXRange(600);
            SnmpFilledGraph1.setXScalePoints(10);
            SnmpFilledGraph1.setXGrids(10);
            SnmpFilledGraph1.setNoOfValues(300);
            SnmpFilledGraph1.setObjectID(".1.3.6.1.2.1.6.10.0");
        }catch(Exception ex){};

        try
        {
            SnmpFilledGraph2.setYGrids(5);
            SnmpFilledGraph2.setAbsoluteTime(true);
            SnmpFilledGraph2.setTargetHost("150.162.252.20");
            SnmpFilledGraph2.setCommunity("painel");
            SnmpFilledGraph2.setXLabel("Tempo");
            SnmpFilledGraph2.setTitle("out Packets");
            SnmpFilledGraph2.setXRange(600);
            SnmpFilledGraph2.setXScalePoints(10);
            SnmpFilledGraph2.setXGrids(10);
            SnmpFilledGraph2.setPollInterval(2);
            SnmpFilledGraph2.setNoOfValues(300);
            SnmpFilledGraph2.setObjectID(".1.3.6.1.2.1.6.11.0");
        }catch(Exception ex){};
    }
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph1 = null;
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph2 = null;
}
```

Anexo J: Código HTML da página IBM8265PacotesTCPDia.html

```
<html>

<head>
<title>SIGMA/WS - ibm8265</title>
</head>

<body>

<p align="center">
<B>Monitoração de Segmentos TCP (24 Horas)</B>
<OBJECT  classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"  width="780"  height="412"
align="baseline"    codebase="http://java.sun.com/products/plugin/1.1/       jinstall-11-
win32.cab #Version=1,1,0,0">
<PARAM NAME="code" VALUE="IBM8265PacotesUDP.class">
<PARAM NAME="codebase" VALUE="../classes">
<PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
<PARAM NAME="archive"VALUE="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<COMMENT>
<EMBED         type="application/x-java-applet"         width="780"         height="412"
code="IBM8265PacotesUDP.class"                                    codebase="../classes"
archive="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<NOEMBED>
</COMMENT>
</NOEMBED>
</EMBED>
</OBJECT>
</p>

</body>

</html>
```

Anexo J1: Código JAVA do applet IBM8265PacotesTCP.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class IBM8265PacotesUDP extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(780,412);
        initialized = true;

        SnmpFilledGraph2= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph2);
        SnmpFilledGraph2.setBounds(10,200,760,200);

        SnmpFilledGraph1= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph1);
        SnmpFilledGraph1.setBounds(10,10,760,190);

        try
        {
            SnmpFilledGraph2.setXScalePoints(6);
            SnmpFilledGraph2.setXGrids(24);
            SnmpFilledGraph2.setYGrids(5);
            SnmpFilledGraph2.setXRange(86400);
            SnmpFilledGraph2.setAbsoluteTime(true);
            SnmpFilledGraph2.setTargetHost("150.162.252.20");
            SnmpFilledGraph2.setCommunity("painel");
            SnmpFilledGraph2.setXLabel("Tempo");
            SnmpFilledGraph2.setNoOfValues(2880);
            SnmpFilledGraph2.setPollInterval(30);
            SnmpFilledGraph2.setTitle("out Packets");
            SnmpFilledGraph2.setObjectID(".1.3.6.1.2.1.7.4.0");
        }catch(Exception ex){};

        try
        {
            SnmpFilledGraph1.setXScalePoints(6);
            SnmpFilledGraph1.setXGrids(24);
            SnmpFilledGraph1.setYGrids(5);
            SnmpFilledGraph1.setXRange(86400);
            SnmpFilledGraph1.setAbsoluteTime(true);
            SnmpFilledGraph1.setTargetHost("150.162.252.20");
            SnmpFilledGraph1.setCommunity("painel");
            SnmpFilledGraph1.setXLabel("Tempo");
            SnmpFilledGraph1.setNoOfValues(2880);
            SnmpFilledGraph1.setPollInterval(30);
            SnmpFilledGraph1.setTitle("in Packtes");
            SnmpFilledGraph1.setObjectID(".1.3.6.1.2.1.7.1.0");
        }catch(Exception ex){};
    }
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph2 = null;
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph1 = null;
}
```

Anexo K: Código HTML da página IBM8265PacotesUDP.html

```
<html>

<head>
<title>SIGMA/WS - ibm8265</title>
</head>

<body>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="15%" align="center"><b><img src="logo.jpg"></b></td>
    <td width="85%" align="center"><p align="center" style="word-spacing: 0; line-
height: 100%; text-indent: 0; margin-left: 0; margin-right: 0; margin-top: 8; margin-
bottom: 0"><b><font size="4">SIGMA/WS - Ferramenta para Gerência de redes ATM, via WWW,
Java e SNMP</font></b></p></td>
  </tr>
</table>
<p align="center"><img border="0" src="ibm8265v.jpg" width="125" height="172"></p>
<p align="center"><b><font size="5">Informações sobre o Tráfego de Datagramas
UDP</font></b></p>

<p align="center">
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="778" height="413"
align="baseline" codebase="http://java.sun.com/products/plugin/1.1/ jinstall-11-
win32.cab #Version=1,1,0,0">
<PARAM NAME="code" VALUE="IBM8265PacUDP.class">
<PARAM NAME="codebase" VALUE="../classes">
<PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
<PARAM NAME="archive"VALUE="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<COMMENT>
<EMBED type="application/x-java-applet" width="778" height="413"
code="IBM8265PacUDP.class" codebase="../classes"
archive="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<NOEMBED>
</COMMENT>
</NOEMBED>
</EMBED>
</OBJECT>

<BR>
<A HREF="http://150.162.60.250:9191/mestrado/html/IBM8265PacotesUDPDia.html"
target=_blank width=400 height=400>Monitoração de Datagramas UDP (24 Horas)</A>

</p>

</body>

</html>
```

Anexo K1: Código JAVA do applet IBM8265PacUDP.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class IBM8265PacUDP extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(778,413);
        initialized = true;

        SnmpFilledGraph1= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph1);
        SnmpFilledGraph1.setBounds(10,10,760,190);

        SnmpFilledGraph2= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph2);
        SnmpFilledGraph2.setBounds(10,200,760,200);

        try
        {
            SnmpFilledGraph1.setYGrids(5);
            SnmpFilledGraph1.setAbsoluteTime(true);
            SnmpFilledGraph1.setTargetHost("150.162.252.20");
            SnmpFilledGraph1.setCommunity("painel");
            SnmpFilledGraph1.setXLabel("Tempo");
            SnmpFilledGraph1.setTitle("in Packtes");
            SnmpFilledGraph1.setObjectID(".1.3.6.1.2.1.7.1.0");
            SnmpFilledGraph1.setXRange(600);
            SnmpFilledGraph1.setXScalePoints(10);
            SnmpFilledGraph1.setXGrids(10);
            SnmpFilledGraph1.setPollInterval(2);
            SnmpFilledGraph1.setNoOfValues(300);
        }catch(Exception ex){};

        try
        {
            SnmpFilledGraph2.setYGrids(5);
            SnmpFilledGraph2.setAbsoluteTime(true);
            SnmpFilledGraph2.setTargetHost("150.162.252.20");
            SnmpFilledGraph2.setCommunity("painel");
            SnmpFilledGraph2.setXLabel("Tempo");
            SnmpFilledGraph2.setTitle("out Packets");
            SnmpFilledGraph2.setObjectID(".1.3.6.1.2.1.7.4.0");
            SnmpFilledGraph2.setXScalePoints(10);
            SnmpFilledGraph2.setXGrids(10);
            SnmpFilledGraph2.setXRange(600);
            SnmpFilledGraph2.setPollInterval(2);
            SnmpFilledGraph2.setNoOfValues(300);
        }catch(Exception ex){};
    }
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph1 = null;
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph2 = null;
}
```

Anexo L: Código HTML da página IBM8265PacotesUDPDia.html

```
<html>

<head>
<title>SIGMA/WS - ibm8265</title>
</head>

<body>

<p align="center">
<B>Monitoração de Datagramas UDP (24 Horas)</B>
<OBJECT  classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"  width="783"  height="405"
align="baseline"     codebase="http://java.sun.com/products/plugin/1.1/     jinstall-11-
win32.cab #Version=1,1,0,0">
<PARAM NAME="code" VALUE="IBM8265PacotesUDP.class">
<PARAM NAME="codebase" VALUE="../classes">
<PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
<PARAM NAME="archive"VALUE="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<COMMENT>
<EMBED         type="application/x-java-applet"         width="783"         height="405"
code="IBM8265PacotesUDP.class"                                    codebase="../classes"
archive="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<NOEMBED>
</COMMENT>
</NOEMBED>
</EMBED>
</OBJECT>
</p>

</body>

</html>
```

Anexo L1: Código JAVA do applet IBM8265PacotesUDP.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class IBM8265PacotesUDP extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(780,412);
        initialized = true;

        SnmpFilledGraph2= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph2);
        SnmpFilledGraph2.setBounds(10,200,760,200);

        SnmpFilledGraph1= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph1);
        SnmpFilledGraph1.setBounds(10,10,760,190);

        try
        {
            SnmpFilledGraph2.setXScalePoints(6);
            SnmpFilledGraph2.setXGrids(24);
            SnmpFilledGraph2.setYGrids(5);
            SnmpFilledGraph2.setXRange(86400);
            SnmpFilledGraph2.setAbsoluteTime(true);
            SnmpFilledGraph2.setTargetHost("150.162.252.20");
            SnmpFilledGraph2.setCommunity("painel");
            SnmpFilledGraph2.setXLabel("Tempo");
            SnmpFilledGraph2.setNoOfValues(2880);
            SnmpFilledGraph2.setPollInterval(30);
            SnmpFilledGraph2.setTitle("out Packets");
            SnmpFilledGraph2.setObjectID(".1.3.6.1.2.1.7.4.0");
       }catch(Exception ex){};

        try
        {
            SnmpFilledGraph1.setXScalePoints(6);
            SnmpFilledGraph1.setXGrids(24);
            SnmpFilledGraph1.setYGrids(5);
            SnmpFilledGraph1.setXRange(86400);
            SnmpFilledGraph1.setAbsoluteTime(true);
            SnmpFilledGraph1.setTargetHost("150.162.252.20");
            SnmpFilledGraph1.setCommunity("painel");
            SnmpFilledGraph1.setXLabel("Tempo");
            SnmpFilledGraph1.setNoOfValues(2880);
            SnmpFilledGraph1.setPollInterval(30);
            SnmpFilledGraph1.setTitle("in Packtes");
            SnmpFilledGraph1.setObjectID(".1.3.6.1.2.1.7.1.0");
        }catch(Exception ex){};
    }
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph2 = null;
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph1 = null;
}
```

Anexo M: Código HTML da página RedeUFSC.html

```
<html>

<head>
<title>SIGMA/WS - redeUFSC</title>
</head>

<body>

<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="15%" align="center"><b><img src="logo.jpg"></b></td>
    <td width="85%" align="center"><p align="center" style="word-spacing: 0; line-
height: 100%; text-indent: 0; margin-left: 0; margin-right: 0; margin-top: 8; margin-
bottom: 0"><b><font size="4">SIGMA/WS - Ferramenta para Gerência de redes ATM, via WWW,
Java e SNMP</font></b></p></td>
  </tr>
</table>
<table border="0" width="100%" cellpadding="0" cellspacing="0">
  <tr>
    <td width="33%">                                    <img border="0" src="redeUFSC.jpg"
align="right" width="104" height="89"></td>
    <td width="34%"><p align="center"><img border="0" src="interfaceLeft.jpg"
width="210" height="71"></td>
    <td width="33%">                                    <img border="0" src="ibm8265vd.jpg"
width="160" height="174"></td>
  </tr>
</table>

<p align="center" style="word-spacing: 0; line-height: 100%; text-indent: 0; margin-
left: 0; margin-top: 0; margin-bottom: 0"> </p>
<p align="center">
 <OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="370" height="210"
align="baseline"     codebase="http://java.sun.com/products/plugin/1.1/     jinstall-11-
win32.cab #Version=1,1,0,0">
 <PARAM NAME="code" VALUE="RedeUFSCStatus.class">
 <PARAM NAME="codebase" VALUE="../classes">
 <PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
 <PARAM NAME="archive"VALUE="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
 <COMMENT>
 <EMBED         type="application/x-java-applet"         width="370"         height="210"
code="RedeUFSCStatus.class"                                    codebase="../classes"
archive="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
 <NOEMBED>
 </COMMENT>
 </NOEMBED>
 </EMBED>
 </OBJECT>
</p>

<p   style="word-spacing:   0;   line-height:   100%;   text-indent:   0;   margin:   0"
align="center"><a href="redeUFSCVolume.html">Volume</a></p>
<p   style="word-spacing:   0;   line-height:   100%;   text-indent:   0;   margin:   0"
align="center"><a href="redeUFSCVazao.html">Vazão</a></p>
<p   style="word-spacing:   0;   line-height:   100%;   text-indent:   0;   margin:   0"
align="center"><a href="redeUFSCQoS.html">QoS</a></p>

</body>

</html>
```

Anexo M1: Código JAVA do applet RedeUFSCStatus.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class RedeUFSCStatus extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(370,210);
        initialized = true;

        JLabel1= new javax.swing.JLabel();
        getContentPane().add(JLabel1);
        JLabel1.setBounds(10,15,160,30);

        JLabel2= new javax.swing.JLabel();
        getContentPane().add(JLabel2);
        JLabel2.setBounds(10,45,160,30);

        JLabel3= new javax.swing.JLabel();
        getContentPane().add(JLabel3);
        JLabel3.setBounds(10,75,160,30);

        JLabel6= new javax.swing.JLabel();
        getContentPane().add(JLabel6);
        JLabel6.setBounds(10,165,160,30);

        SnmpLed1= new com.adventnet.netmonitor.SnmpLed(this);
        SnmpLed1.setLayout(null);
        getContentPane().add(SnmpLed1);
        SnmpLed1.setBounds(170,15,190,30);

        SnmpDigDisp3= new com.adventnet.netmonitor.SnmpDigDisp(this);
        getContentPane().add(SnmpDigDisp3);
        SnmpDigDisp3.setBounds(170,75,190,30);

        SnmpDigDisp4= new com.adventnet.netmonitor.SnmpDigDisp(this);
        getContentPane().add(SnmpDigDisp4);
        SnmpDigDisp4.setBounds(170,105,190,30);

        OutErrorsDigDisp6= new com.adventnet.netmonitor.SnmpDigDisp(this);
        getContentPane().add(OutErrorsDigDisp6);
        OutErrorsDigDisp6.setBounds(170,165,190,30);

        SnmpDigDisp5= new com.adventnet.netmonitor.SnmpDigDisp(this);
        getContentPane().add(SnmpDigDisp5);
        SnmpDigDisp5.setBounds(170,135,190,30);

        JLabel5= new javax.swing.JLabel();
        getContentPane().add(JLabel5);
        JLabel5.setBounds(10,135,160,30);

        SnmpTextField2= new com.adventnet.netmonitor.SnmpTextField(this);
        getContentPane().add(SnmpTextField2);
        SnmpTextField2.setBounds(170,45,190,30);

        JLabel4= new javax.swing.JLabel();
        getContentPane().add(JLabel4);
        JLabel4.setBounds(10,105,160,30);

        try
        {
            JLabel1.setText("OperStatus");
            JLabel1.setHorizontalAlignment(0);
        }catch(Exception ex){};

        try
        {
```

```
    JLabel2.setText("Speed");
    JLabel2.setHorizontalAlignment(0);
}catch(Exception ex){};

try
{
    JLabel3.setHorizontalAlignment(0);
    JLabel3.setText("InOctets");
}catch(Exception ex){};

try
{
    JLabel6.setText("OutErrors");
    JLabel6.setHorizontalAlignment(0);
}catch(Exception ex){};

try
{
    SnmpLed1.setNumberOfStates(3);
    SnmpLed1.setThresholdColor3(new Color(-154));
    SnmpLed1.setThresholdLabel3("Testing");
    SnmpLed1.setShowValue(false);
    SnmpLed1.setThresholdValue1(1);
    SnmpLed1.setThresholdValue2(2);
    SnmpLed1.setThresholdValue3(3);
    SnmpLed1.setThresholdLabel1("Up");
    SnmpLed1.setThresholdLabel2("Down");
    SnmpLed1.setThresholdColor1(new Color(-10027264));
    SnmpLed1.setThresholdColor2(new Color(-65536));
    SnmpLed1.setFgColor(new Color(-1));
    SnmpLed1.setTargetHost("150.162.252.20");
    SnmpLed1.setCommunity("painel");
    SnmpLed1.setPollInterval(10);
    SnmpLed1.setTargetPort(161);
    SnmpLed1.setObjectID(".1.3.6.1.2.1.2.2.1.8.1701");
}catch(Exception ex){};

try
{
    SnmpDigDisp3.setCommunity("painel");
    SnmpDigDisp3.setTargetHost("150.162.252.20");
    SnmpDigDisp3.setObjectID(".1.3.6.1.2.1.2.2.1.10.1701");
    SnmpDigDisp3.setPollInterval(10);
    SnmpDigDisp3.setTargetPort(161);
}catch(Exception ex){};

try
{
    SnmpDigDisp4.setCommunity("painel");
    SnmpDigDisp4.setTargetHost("150.162.252.20");
    SnmpDigDisp4.setPollInterval(10);
    SnmpDigDisp4.setObjectID(".1.3.6.1.2.1.2.2.1.16.1701");
    SnmpDigDisp4.setTargetPort(161);
}catch(Exception ex){};

try
{
    OutErrorsDigDisp6.setForeground(new Color(-65536));
    OutErrorsDigDisp6.setCommunity("painel");
    OutErrorsDigDisp6.setTargetHost("150.162.252.20");
    OutErrorsDigDisp6.setObjectID(".1.3.6.1.2.1.2.2.1.20.1701");
    OutErrorsDigDisp6.setPollInterval(10);
    OutErrorsDigDisp6.setTargetPort(161);
}catch(Exception ex){};

try
{
    SnmpDigDisp5.setForeground(new Color(-65536));
    SnmpDigDisp5.setCommunity("painel");
    SnmpDigDisp5.setTargetHost("150.162.252.20");
    SnmpDigDisp5.setObjectID(".1.3.6.1.2.1.2.2.1.14.1701");
    SnmpDigDisp5.setPollInterval(10);
    SnmpDigDisp5.setTargetPort(161);
}catch(Exception ex){};

try
{
    JLabel5.setText("InErrors");
```

```
            JLabel5.setHorizontalAlignment(0);
        }catch(Exception ex){};

        try
        {
            SnmpTextField2.setCommunity("painel");
            SnmpTextField2.setTargetHost("150.162.252.20");
            SnmpTextField2.setObjectID(".1.3.6.1.2.1.2.2.1.5.1701");
            SnmpTextField2.setBackground(new Color(-16777216));
            SnmpTextField2.setForeground(new Color(-1));
            SnmpTextField2.setPollInterval(10);
        }catch(Exception ex){};

        try
        {
            JLabel4.setText("OutOctets");
            JLabel4.setHorizontalAlignment(0);
        }catch(Exception ex){};
    }
    javax.swing.JLabel  JLabel1 = null;
    javax.swing.JLabel  JLabel2 = null;
    javax.swing.JLabel  JLabel3 = null;
    javax.swing.JLabel  JLabel6 = null;
    com.adventnet.netmonitor.SnmpLed  SnmpLed1 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp3 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp4 = null;
    com.adventnet.netmonitor.SnmpDigDisp  OutErrorsDigDisp6 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp5 = null;
    javax.swing.JLabel  JLabel5 = null;
    com.adventnet.netmonitor.SnmpTextField  SnmpTextField2 = null;
    javax.swing.JLabel  JLabel4 = null;
}
```

Anexo N: Código HTML da página RedeUFSCVolume.html

```
<html>

<head>
<title>SIGMA/WS - redeUFSC</title>
</head>

<body>

<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="15%" align="center"><b><img src="logo.jpg"></b></td>
    <td width="85%" align="center"><p align="center" style="word-spacing: 0; line-
height: 100%; text-indent: 0; margin-left: 0; margin-right: 0; margin-top: 8; margin-
bottom: 0"><b><font size="4">SIGMA/WS - Ferramenta para Gerência de redes ATM, via WWW,
Java e SNMP</font></b></p></td>
  </tr>
</table>
<table border="0" width="100%" cellpadding="0" cellspacing="0">
  <tr>
    <td width="33%">                                    <img border="0" src="redeUFSC.jpg"
align="right" width="104" height="89"></td>
    <td width="34%"><p align="center"><img border="0" src="interfaceLeft.jpg"
width="210" height="71"></td>
    <td width="33%">                                    <img border="0" src="ibm8265vd.jpg"
width="160" height="174"></td>
  </tr>
</table>
<p align="center"><b><font size="5">Informações sobre o Volume</font></b></p>

<p align="center">
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="783" height="420"
align="baseline" codebase="http://java.sun.com/products/plugin/1.1/ jinstall-11-
win32.cab #Version=1,1,0,0">
<PARAM NAME="code" VALUE="redeVol.class">
<PARAM NAME="codebase" VALUE="../classes">
<PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
<PARAM NAME="archive"VALUE="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<COMMENT>
<EMBED type="application/x-java-applet" width="783" height="420" code="redeVol.class"
codebase="../classes" archive="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<NOEMBED>
</COMMENT>
</NOEMBED>
</EMBED>
</OBJECT>

<BR>
<A HREF="http://150.162.60.250:9191/mestrado/html/redeUFSCVolumeDia.html" target=_blank
width=600 height=500>Monitoração de Tráfego na Conexão redeUFSC (24 Horas)</A>

</p>

</body>

</html>
```

Anexo N1: Código JAVA do applet RedeVol.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class redeVol extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(783,420);
        initialized = true;

        SnmpFilledGraph3= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph3);
        SnmpFilledGraph3.setBounds(10,10,760,200);

        SnmpFilledGraph2= new com.adventnet.netmonitor.SnmpFilledGraph(this);
        getContentPane().add(SnmpFilledGraph2);
        SnmpFilledGraph2.setBounds(10,210,760,200);

        try
        {
            SnmpFilledGraph3.setYGrids(5);
            SnmpFilledGraph3.setAbsoluteTime(true);
            SnmpFilledGraph3.setTargetHost("150.162.252.20");
            SnmpFilledGraph3.setCommunity("painel");
            SnmpFilledGraph3.setXLabel("Tempo");
            SnmpFilledGraph3.setXScalePoints(10);
            SnmpFilledGraph3.setXGrids(10);
            SnmpFilledGraph3.setXRange(600);
            SnmpFilledGraph3.setNoOfValues(300);
            SnmpFilledGraph3.setPollInterval(2);
            SnmpFilledGraph3.setObjectID(".1.3.6.1.2.1.2.2.1.10.1701");
            SnmpFilledGraph3.setTitle("in Octets");
        }catch(Exception ex){};

        try
        {
            SnmpFilledGraph2.setYGrids(5);
            SnmpFilledGraph2.setAbsoluteTime(true);
            SnmpFilledGraph2.setTargetHost("150.162.252.20");
            SnmpFilledGraph2.setCommunity("painel");
            SnmpFilledGraph2.setXLabel("Tempo");
            SnmpFilledGraph2.setObjectID(".1.3.6.1.2.1.2.2.1.16.1701");
            SnmpFilledGraph2.setTitle("out Octets");
            SnmpFilledGraph2.setXScalePoints(10);
            SnmpFilledGraph2.setXGrids(10);
            SnmpFilledGraph2.setXRange(600);
            SnmpFilledGraph2.setNoOfValues(300);
            SnmpFilledGraph2.setPollInterval(2);
        }catch(Exception ex){};
    }
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph3 = null;
    com.adventnet.netmonitor.SnmpFilledGraph  SnmpFilledGraph2 = null;
}
```

Anexo O: Código HTML da página RedeUFSCVolumeDia.html

```
<html>

<head>
<title>SIGMA/WS - redeUFSC</title>
</head>

<body>
<p align="center"><b><font size="5">Informações sobre o Volume - Conexão redeUFSC (24
Horas)</font></b></p>
<p align="center">
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="784" height="419"
align="baseline" codebase="http://java.sun.com/products/plugin/1.1/ jinstall-11-
win32.cab #Version=1,1,0,0">
<PARAM NAME="code" VALUE="redeUFSCVolume.class">
<PARAM NAME="codebase" VALUE="../classes">
<PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
<PARAM NAME="archive"VALUE="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<COMMENT>
<EMBED type="application/x-java-applet" width="784" height="419"
code="redeUFSCVolume.class" codebase="../classes"
archive="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
<NOEMBED>
</COMMENT>
</NOEMBED>
</EMBED>
</OBJECT>
</p>

</body>

</html>
```

Anexo O1: Código JAVA do applet RedeUFSCVolume.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class redeUFSCVolume extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(784,407);
        initialized = true;

        LineGraph1= new com.adventnet.beans.graphs.LineGraph();
        getContentPane().add(LineGraph1);
        LineGraph1.setBounds(10,10,765,190);

        LineGraph2= new com.adventnet.beans.graphs.LineGraph();
        getContentPane().add(LineGraph2);
        LineGraph2.setBounds(10,200,765,195);

        SnmpPoller1= new com.adventnet.snmp.beans.SnmpPoller(this);

        try
        {
            LineGraph1.setAbsoluteTime(true);
            LineGraph1.setXGrids(24);
            LineGraph1.setXScalePoints(6);
            LineGraph1.setXLabel("Tempo");
            LineGraph1.setNoOfValues(288);
            LineGraph1.setXRange(86400);
            LineGraph1.setTitle("in Octets");
            LineGraph1.setMaxY(155000);
        }catch(Exception ex){};

        try
        {
            LineGraph2.setAbsoluteTime(true);
            LineGraph2.setXScalePoints(6);
            LineGraph2.setXGrids(24);
            LineGraph2.setXLabel("Tempo");
            LineGraph2.setXRange(86400);
            LineGraph2.setNoOfValues(288);
            LineGraph2.setMaxY(155000);
            LineGraph2.setTitle("out Octets");
        }catch(Exception ex){};

        try
        {
            SnmpPoller1.setTargetHost("150.162.252.20");
            SnmpPoller1.setTargetPort(161);
            SnmpPoller1.setTimeout(1);
            SnmpPoller1.setCommunity("painel");
            SnmpPoller1.setObjectID(".1.3.6.1.2.1.2.2.1.10.1501");
            java.lang.String[]  objectIDList_array = new java.lang.String[ 2 ];
            objectIDList_array[ 0 ] = ".1.3.6.1.2.1.2.2.1.10.1701";
            objectIDList_array[ 1 ] = ".1.3.6.1.2.1.2.2.1.16.1701";
            SnmpPoller1.setObjectIDList(objectIDList_array);
            SnmpPoller1.setPollInterval(300);
        }catch(Exception ex){};

      SnmpPoller1_LineGraph1_conn4      SnmpPoller1_LineGraph1_conn41      =            new
SnmpPoller1_LineGraph1_conn4();
      SnmpPoller1_LineGraph1_conn41.setTarget( LineGraph1);
      SnmpPoller1.addResultListener(SnmpPoller1_LineGraph1_conn41);
      SnmpPoller1_LineGraph2_conn5      SnmpPoller1_LineGraph2_conn51      =            new
SnmpPoller1_LineGraph2_conn5();
      SnmpPoller1_LineGraph2_conn51.setTarget( LineGraph2);
      SnmpPoller1.addResultListener(SnmpPoller1_LineGraph2_conn51);
    }
```

```
    com.adventnet.beans.graphs.LineGraph  LineGraph1 = null;
    com.adventnet.beans.graphs.LineGraph  LineGraph2 = null;
    com.adventnet.snmp.beans.SnmpPoller  SnmpPoller1 = null;
}
```

Anexo P: Código HTML da página RedeUFSCVazao.html

```
<html>

<head>
<title>SIGMA/WS - redeUFSC</title>
</head>

<body>
<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="15%" align="center"><b><img src="logo.jpg"></b></td>
    <td width="85%" align="center"><p align="center" style="word-spacing: 0; line-
height: 100%; text-indent: 0; margin-left: 0; margin-right: 0; margin-top: 8; margin-
bottom: 0"><b><font size="4">SIGMA/WS - Ferramenta para Gerência de redes ATM, via WWW,
Java e SNMP</font></b></p></td>
  </tr>
</table>
<table border="0" width="100%" cellpadding="0" cellspacing="0">
  <tr>
    <td width="33%">                                    <img border="0" src="redeUFSC.jpg"
align="right" width="104" height="89"></td>
    <td width="34%"><p align="center"><img border="0" src="interfaceLeft.jpg"
width="210" height="71"></td>
    <td width="33%">                                    <img border="0" src="ibm8265vd.jpg"
width="160" height="174"></td>
  </tr>
</table>
<p align="center"><b><font size="5">Informações sobre a Vazão</font></b></p>

<p align="center">
 <OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="300" height="316"
align="baseline" codebase="http://java.sun.com/products/plugin/1.1/ jinstall-11-
win32.cab #Version=1,1,0,0">
 <PARAM NAME="code" VALUE="redeUFSCVazao.class">
 <PARAM NAME="codebase" VALUE="../classes">
 <PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
 <PARAM
NAME="archive"VALUE="Containers.jar,BuilderSwing.jar,AdventNetSnmp.jar,AdventNetUI.jar,N
etMonitor.jar">
 <COMMENT>
 <EMBED type="application/x-java-applet" width="300" height="316"
code="redeUFSCVazao.class" codebase="../classes"
archive="Containers.jar,BuilderSwing.jar,AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.ja
r">
 <NOEMBED>
 </COMMENT>
 </NOEMBED>
 </EMBED>
 </OBJECT>

 <BR>
 <A HREF="http://150.162.60.250:9191/mestrado/html/redeUFSCVazaoDia.html" target=_blank
width=600 height=700>Monitoração da Vazão na Conexão redeUFSC (24 Horas)</A>

</p>

</body>

</html>
```

Anexo P1: Código JAVA do applet RedeUFSCVazao.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class redeUFSCVazao extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(300,316);
        initialized = true;

        SnmpVCellularGauge1= new com.adventnet.netmonitor.SnmpVCellularGauge(this);
        SnmpVCellularGauge1.setLayout(null);
        getContentPane().add(SnmpVCellularGauge1);
        SnmpVCellularGauge1.setBounds(10,30,135,275);

        JLabel3= new com.adventnet.swing.JLabel();
        getContentPane().add(JLabel3);
        JLabel3.setBounds(10,10,135,20);

        SnmpVCellularGauge2= new com.adventnet.netmonitor.SnmpVCellularGauge(this);
        SnmpVCellularGauge2.setLayout(null);
        getContentPane().add(SnmpVCellularGauge2);
        SnmpVCellularGauge2.setBounds(155,30,135,275);

        JLabel4= new com.adventnet.swing.JLabel();
        getContentPane().add(JLabel4);
        JLabel4.setBounds(155,10,135,20);

        try
        {
            SnmpVCellularGauge1.setTargetHost("150.162.252.20");
            SnmpVCellularGauge1.setCommunity("painel");
            SnmpVCellularGauge1.setObjectID(".1.3.6.1.2.1.2.2.1.10.1701");
            SnmpVCellularGauge1.setPollInterval(2);
            SnmpVCellularGauge1.setMarkThreshold(true);
            SnmpVCellularGauge1.setMaxValue(385000);
            SnmpVCellularGauge1.setThresholdValue1(192500);
            SnmpVCellularGauge1.setThresholdValue2(288750);
        }catch(Exception ex){};

        try
        {
            JLabel3.setText("in");
            JLabel3.setHorizontalAlignment(0);
        }catch(Exception ex){};

        try
        {
            SnmpVCellularGauge2.setTargetHost("150.162.252.20");
            SnmpVCellularGauge2.setPollInterval(2);
            SnmpVCellularGauge2.setCommunity("painel");
            SnmpVCellularGauge2.setObjectID(".1.3.6.1.2.1.2.2.1.16.1701");
            SnmpVCellularGauge2.setMaxValue(385000);
            SnmpVCellularGauge2.setThresholdValue1(192500);
            SnmpVCellularGauge2.setThresholdValue2(288750);
        }catch(Exception ex){};

        try
        {
            JLabel4.setHorizontalAlignment(0);
            JLabel4.setText("out");
        }catch(Exception ex){};
    }
    com.adventnet.netmonitor.SnmpVCellularGauge  SnmpVCellularGauge1 = null;
    com.adventnet.swing.JLabel  JLabel3 = null;
    com.adventnet.netmonitor.SnmpVCellularGauge  SnmpVCellularGauge2 = null;
```

```
    com.adventnet.swing.JLabel  JLabel4 = null;
}
```

Anexo Q: Código HTML da página RedeUFSCVazaoDia.html

```
<html>

<head>
<title>SIGMA/WS - redeUFSC</title>
</head>

<body>

<p align="center"><b><font size="5">Informações sobre a Vazão - Conexão redeUFSC (24
Horas)</font></b></p>

<p align="center">
 <OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" width="760" height="542"
align="baseline"     codebase="http://java.sun.com/products/plugin/1.1/     jinstall-11-
win32.cab #Version=1,1,0,0">
 <PARAM NAME="code" VALUE="redeUFSCVazaoDia.class">
 <PARAM NAME="codebase" VALUE="../classes">
 <PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
 <PARAM NAME="archive"VALUE="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
 <COMMENT>
 <EMBED         type="application/x-java-applet"         width="760"         height="542"
code="redeUFSCVazaoDia.class"                                    codebase="../classes"
archive="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
 <NOEMBED>
 </COMMENT>
 </NOEMBED>
 </EMBED>
 </OBJECT>
</p>

</body>

</html>
```

Anexo Q1: Código HTML da página RedeUFSCVazaoDia.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class redeUFSCVazaoDia extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(760,542);
        initialized = true;

        SnmpLineGraph1= new com.adventnet.netmonitor.SnmpLineGraph(this);
        getContentPane().add(SnmpLineGraph1);
        SnmpLineGraph1.setBounds(10,10,740,260);

        SnmpLineGraph2= new com.adventnet.netmonitor.SnmpLineGraph(this);
        getContentPane().add(SnmpLineGraph2);
        SnmpLineGraph2.setBounds(10,270,740,260);

        try
        {
            SnmpLineGraph1.setTargetHost("150.162.252.20");
            SnmpLineGraph1.setCommunity("painel");
            SnmpLineGraph1.setObjectID(".1.3.6.1.2.1.2.2.1.10.1701");
            SnmpLineGraph1.setXScalePoints(6);
            SnmpLineGraph1.setXGrids(24);
            SnmpLineGraph1.setTitle("ifInOctets");
            SnmpLineGraph1.setAbsoluteTime(true);
            SnmpLineGraph1.setXRange(86400);
            SnmpLineGraph1.setNoOfValues(2880);
            SnmpLineGraph1.setPollInterval(30);
            SnmpLineGraph1.setYGrids(5);
            SnmpLineGraph1.setYScalePoints(5);
            SnmpLineGraph1.setXLabel("Tempo");
            SnmpLineGraph1.setMaxY(5775000);
        }catch(Exception ex){};

        try
        {
            SnmpLineGraph2.setTargetHost("150.162.252.20");
            SnmpLineGraph2.setCommunity("painel");
            SnmpLineGraph2.setXScalePoints(6);
            SnmpLineGraph2.setXGrids(24);
            SnmpLineGraph2.setAbsoluteTime(true);
            SnmpLineGraph2.setXRange(86400);
            SnmpLineGraph2.setNoOfValues(2880);
            SnmpLineGraph2.setPollInterval(30);
            SnmpLineGraph2.setYGrids(5);
            SnmpLineGraph2.setYScalePoints(5);
            SnmpLineGraph2.setTitle("ifOutOctets");
            SnmpLineGraph2.setObjectID(".1.3.6.1.2.1.2.2.1.16.1701");
            SnmpLineGraph2.setXLabel("Tempo");
            SnmpLineGraph2.setMaxY(5775000);
        }catch(Exception ex){};
    }
    com.adventnet.netmonitor.SnmpLineGraph  SnmpLineGraph1 = null;
    com.adventnet.netmonitor.SnmpLineGraph  SnmpLineGraph2 = null;
}
```

Anexo R: Código HTML da página RedeUFSCQoS.html

```
<html>

<head>
<title>SIGMA/WS - redeUFSC</title>
</head>

<body>

<table border="0" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td width="15%" align="center"><b><img src="logo.jpg"></b></td>
    <td  width="85%"  align="center"><p  align="center"  style="word-spacing:  0;  line-
height: 100%; text-indent: 0; margin-left: 0; margin-right: 0; margin-top: 8; margin-
bottom: 0"><b><font size="4">SIGMA/WS - Ferramenta para Gerência de redes ATM, via WWW,
Java e SNMP</font></b></p></td>
  </tr>
</table>
<table border="0" width="100%" cellpadding="0" cellspacing="0">
  <tr>
    <td  width="33%">                                    <img  border="0"  src="redeUFSC.jpg"
align="right" width="104" height="89"></td>
    <td    width="34%"><p    align="center"><img    border="0"    src="interfaceLeft.jpg"
width="210" height="71"></td>
    <td  width="33%">                                   <img  border="0"  src="ibm8265vd.jpg"
width="160" height="174"></td>
  </tr>
</table>
<p align="center"><b><font size="5">Informações sobre a redeUFSC</font></b></p>

<p align="center">
 <OBJECT  classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"  width="771"  height="492"
align="baseline"    codebase="http://java.sun.com/products/plugin/1.1/       jinstall-11-
win32.cab #Version=1,1,0,0">
 <PARAM NAME="code" VALUE="redeUFSCQoS.class">
 <PARAM NAME="codebase" VALUE="../classes">
 <PARAM NAME="type"VALUE="application/x-java-applet;version=1.1">
 <PARAM NAME="archive"VALUE="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
 <COMMENT>
 <EMBED         type="application/x-java-applet"         width="771"         height="492"
code="redeUFSCQoS.class"                                          codebase="../classes"
archive="AdventNetSnmp.jar,AdventNetUI.jar,NetMonitor.jar">
 <NOEMBED>
 </COMMENT>
 </NOEMBED>
 </EMBED>
 </OBJECT>
</p>

</body>

</html>
```

Anexo R1: Código JAVA do applet RedeUFSCQoS.java

```
import javax.swing.*;
import java.io.*;
import java.awt.*;

public class redeUFSCQoS extends  JApplet
{
    private boolean initialized = false;
    public void init()
    {
        try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
            } catch (Exception e) {}
        if (initialized == true) return;
        getContentPane().setLayout(null);
        setSize(771,492);
        initialized = true;

        JLabel1= new javax.swing.JLabel();
        getContentPane().add(JLabel1);
        JLabel1.setBounds(10,10,135,30);

        SnmpDigDisp1= new com.adventnet.netmonitor.SnmpDigDisp(this);
        getContentPane().add(SnmpDigDisp1);
        SnmpDigDisp1.setBounds(145,10,135,30);

        JLabel2= new javax.swing.JLabel();
        getContentPane().add(JLabel2);
        JLabel2.setBounds(10,55,135,30);

        JLabel3= new javax.swing.JLabel();
        getContentPane().add(JLabel3);
        JLabel3.setBounds(10,85,135,30);

        JLabel4= new javax.swing.JLabel();
        getContentPane().add(JLabel4);
        JLabel4.setBounds(10,115,135,30);

        SnmpDigDisp4= new com.adventnet.netmonitor.SnmpDigDisp(this);
        getContentPane().add(SnmpDigDisp4);
        SnmpDigDisp4.setBounds(145,115,135,30);

        SnmpDigDisp5= new com.adventnet.netmonitor.SnmpDigDisp(this);
        getContentPane().add(SnmpDigDisp5);
        SnmpDigDisp5.setBounds(145,145,135,30);

        JLabel5= new javax.swing.JLabel();
        getContentPane().add(JLabel5);
        JLabel5.setBounds(10,145,135,30);

        JLabel8= new javax.swing.JLabel();
        getContentPane().add(JLabel8);
        JLabel8.setBounds(10,285,135,30);

        JLabel9= new javax.swing.JLabel();
        getContentPane().add(JLabel9);
        JLabel9.setBounds(10,315,135,30);

        SnmpDigDisp9= new com.adventnet.netmonitor.SnmpDigDisp(this);
        getContentPane().add(SnmpDigDisp9);
        SnmpDigDisp9.setBounds(145,315,135,30);

        JLabel10= new javax.swing.JLabel();
        getContentPane().add(JLabel10);
        JLabel10.setBounds(10,345,135,30);

        SnmpDigDisp10= new com.adventnet.netmonitor.SnmpDigDisp(this);
        getContentPane().add(SnmpDigDisp10);
        SnmpDigDisp10.setBounds(145,345,135,30);

        SnmpMultiLineGraph2= new com.adventnet.netmonitor.SnmpMultiLineGraph(this);
        getContentPane().add(SnmpMultiLineGraph2);
        SnmpMultiLineGraph2.setBounds(280,285,485,195);

        SnmpMultiLineGraph1= new com.adventnet.netmonitor.SnmpMultiLineGraph(this);
```

```
getContentPane().add(SnmpMultiLineGraph1);
SnmpMultiLineGraph1.setBounds(280,55,485,220);

SnmpDigDisp3= new com.adventnet.netmonitor.SnmpDigDisp(this);
getContentPane().add(SnmpDigDisp3);
SnmpDigDisp3.setBounds(145,85,135,30);

SnmpDigDisp15= new com.adventnet.netmonitor.SnmpDigDisp(this);
getContentPane().add(SnmpDigDisp15);
SnmpDigDisp15.setBounds(145,55,135,30);

SnmpDigDisp16= new com.adventnet.netmonitor.SnmpDigDisp(this);
getContentPane().add(SnmpDigDisp16);
SnmpDigDisp16.setBounds(145,285,135,30);

try
{
    JLabel1.setText("CPUAverageLoad");
}catch(Exception ex){};

try
{
    SnmpDigDisp1.setCommunity("painel");
    SnmpDigDisp1.setTargetHost("150.162.252.20");
    SnmpDigDisp1.setObjectID(".1.3.6.1.4.1.2.6.33.1.8.1.1.0");
    SnmpDigDisp1.setPollInterval(10);
    SnmpDigDisp1.setForeground(new Color(-5263441));
}catch(Exception ex){};

try
{
    JLabel2.setText("InOctets");
}catch(Exception ex){};

try
{
    JLabel3.setText("IFinDiscards");
}catch(Exception ex){};

try
{
    JLabel4.setText("IFinErros");
}catch(Exception ex){};

try
{
    SnmpDigDisp4.setCommunity("painel");
    SnmpDigDisp4.setTargetHost("150.162.252.20");
    SnmpDigDisp4.setPollInterval(20);
    SnmpDigDisp4.setForeground(new Color(-65536));
    SnmpDigDisp4.setObjectID(".1.3.6.1.2.1.2.2.1.14.1701");
}catch(Exception ex){};

try
{
    SnmpDigDisp5.setCommunity("painel");
    SnmpDigDisp5.setTargetHost("150.162.252.20");
    SnmpDigDisp5.setPollInterval(20);
    SnmpDigDisp5.setForeground(new Color(-16776961));
    SnmpDigDisp5.setObjectID(".1.3.6:1.2.1.2.2.1.15.1701");
}catch(Exception ex){};

try
{
    JLabel5.setText("IFinUnknowProtos");
}catch(Exception ex){};

try
{
    JLabel8.setText("IPOutRequest");
}catch(Exception ex){};

try
{
    JLabel9.setText("IPOutDiscards");
}catch(Exception ex){};

try
```

```
{
    SnmpDigDisp9.setCommunity("painel");
    SnmpDigDisp9.setTargetHost("150.162.252.20");
    SnmpDigDisp9.setPollInterval(10);
    SnmpDigDisp9.setForeground(new Color(-392961));
    SnmpDigDisp9.setObjectID(".1.3.6.1.2.1.2.2.1.19.1701");
}catch(Exception ex){};

try
{
    JLabel10.setText("IPOutErrors");
}catch(Exception ex){};

try
{
    SnmpDigDisp10.setCommunity("painel");
    SnmpDigDisp10.setTargetHost("150.162.252.20");
    SnmpDigDisp10.setPollInterval(10);
    SnmpDigDisp10.setForeground(new Color(-65536));
    SnmpDigDisp10.setObjectID(".1.3.6.1.2.1.2.2.1.20.1701");
}catch(Exception ex){};

try
{
    SnmpMultiLineGraph2.setCommunity("painel");
    SnmpMultiLineGraph2.setTargetHost("150.162.252.20");
    SnmpMultiLineGraph2.setXScalePoints(10);
    SnmpMultiLineGraph2.setXGrids(10);
    SnmpMultiLineGraph2.setXRange(600);
    SnmpMultiLineGraph2.setNoOfValues(300);
    SnmpMultiLineGraph2.setXLabel("Tempo");
    SnmpMultiLineGraph2.setThresholdValue(50);
    SnmpMultiLineGraph2.setAbsoluteTime(true);
    SnmpMultiLineGraph2.setMaxY(30);
    SnmpMultiLineGraph2.setPollInterval(2);
    SnmpMultiLineGraph2.setTitle("Out");
    java.lang.String[]  Object_ID_array = new java.lang.String[ 3 ];
    Object_ID_array[ 0 ] = ".1.3.6.1.2.1.2.2.1.16.1701";
    Object_ID_array[ 1 ] = ".1.3.6.1.2.1.2.2.1.19.1701";
    Object_ID_array[ 2 ] = ".1.3.6.1.2.1.2.2.1.20.1701";
    SnmpMultiLineGraph2.setObjectIDList(Object_ID_array);
}catch(Exception ex){};

try
{
    SnmpMultiLineGraph1.setCommunity("painel");
    SnmpMultiLineGraph1.setTargetHost("150.162.252.20");
    SnmpMultiLineGraph1.setXScalePoints(10);
    SnmpMultiLineGraph1.setXGrids(10);
    SnmpMultiLineGraph1.setXRange(600);
    SnmpMultiLineGraph1.setNoOfValues(300);
    SnmpMultiLineGraph1.setXLabel("Tempo");
    SnmpMultiLineGraph1.setThresholdValue(50);
    SnmpMultiLineGraph1.setAbsoluteTime(true);
    SnmpMultiLineGraph1.setMaxY(30);
    SnmpMultiLineGraph1.setPollInterval(2);
    SnmpMultiLineGraph1.setTitle("In");
    java.lang.String[]  Object_ID_array = new java.lang.String[ 4 ];
    Object_ID_array[ 0 ] = ".1.3.6.1.2.1.2.2.1.10.1701";
    Object_ID_array[ 1 ] = ".1.3.6.1.2.1.2.2.1.13.1701";
    Object_ID_array[ 2 ] = ".1.3.6.1.2.1.2.2.1.14.1701";
    Object_ID_array[ 3 ] = ".1.3.6.1.2.1.2.2.1.15.1701";
    SnmpMultiLineGraph1.setObjectIDList(Object_ID_array);
}catch(Exception ex){};

try
{
    SnmpDigDisp3.setCommunity("painel");
    SnmpDigDisp3.setTargetHost("150.162.252.20");
    SnmpDigDisp3.setPollInterval(20);
    SnmpDigDisp3.setForeground(new Color(-65281));
    SnmpDigDisp3.setObjectID(".1.3.6.1.2.1.2.2.1.13.1701");
}catch(Exception ex){};

try
{
    SnmpDigDisp15.setCommunity("painel");
    SnmpDigDisp15.setTargetHost("150.162.252.20");
```

```
            SnmpDigDisp15.setPollInterval(10);
            SnmpDigDisp15.setForeground(new Color(-1));
            SnmpDigDisp15.setObjectID(".1.3.6.1.2.1.2.2.1.10.1701");
        }catch(Exception ex){};

        try
        {
            SnmpDigDisp16.setCommunity("painel");
            SnmpDigDisp16.setTargetHost("150.162.252.20");
            SnmpDigDisp16.setPollInterval(10);
            SnmpDigDisp16.setForeground(new Color(-1));
            SnmpDigDisp16.setObjectID(".1.3.6.1.2.1.2.2.1.16.1701");
        }catch(Exception ex){};
    }
    javax.swing.JLabel  JLabel1 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp1 = null;
    javax.swing.JLabel  JLabel2 = null;
    javax.swing.JLabel  JLabel3 = null;
    javax.swing.JLabel  JLabel4 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp4 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp5 = null;
    javax.swing.JLabel  JLabel5 = null;
    javax.swing.JLabel  JLabel8 = null;
    javax.swing.JLabel  JLabel9 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp9 = null;
    javax.swing.JLabel  JLabel10 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp10 = null;
    com.adventnet.netmonitor.SnmpMultiLineGraph  SnmpMultiLineGraph2 = null;
    com.adventnet.netmonitor.SnmpMultiLineGraph  SnmpMultiLineGraph1 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp3 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp15 = null;
    com.adventnet.netmonitor.SnmpDigDisp  SnmpDigDisp16 = null;
}
```

Capítulo 09: Bibliografia

Capítulo 09. Bibliografia

SOARES, L. F. G.: *Redes de Computadores: das LAN's, MAN's, WAN's às redes ATM.* Rio de Janeiro: Campus, 1997

TANENBAUM, A. S.: *Redes de Computadores*; tradução [da 3. ed. Original]. Rio de Janeiro: Campus, 1997

BAROTTO, A. M.: *Realização da Gerência Distribuída de Redes utilizando SNMP, Java, WWW e CORBA;* Dissertação de Mestrado em Ciência da Computação – UFSC. Florianópolis: 1998.

CERUTTI, F. A.: *Gerência de Tráfego em Redes ATM utilizando o protocolo SNMP e a tecnologia Web;* Trabalho individual para encaminhamento de Dissertação de Mestrado em Ciência da Computação – UFSC. Florianópolis: 1998.

SANTOS, Marcílio D. dos: *Entrevista ao jornalista Rogério Mosimann;* Jornal O Estado OnLine: Novembro/1996.

ALLES, Anthony: *ATM Internetworking;* Engineering InterOp. Las Vegas: 1995.

ALVES, Luiz: *Comunicação de Dados;* 2ª Edição Revisada e Ampliada; Makron Books, São Paulo: 1994.

WIRTH, A.: *Redes Digitais de Serviços;* 1ª. Edição; Book Express, Rio de Janeiro: 1998.

KRISHNAN, Kris. FULLER Wayne: *An Overview of MIB's for ATM Network Management;* Volume 5, Número 4; The ATM Forum Newsletter; ATM Forum: Dezembro/1997.

HYDE, Douglas: *Web-Based Management: The new paradigm for network Management;* 3COM Networking Solutions Center: http://www.3com/technology/tech_net/white_papers/50062.html. 1998.

ALEXANDER, Peter. CARPENTER, Kacey: *ATM Net Management: A Status Report;* Data Communications: Setembro/1995.

MODARRES, Houman: *ATM rises to the Challenge;* Volume 5 Número 4; The ATM Forum Newsletter; ATM Forum: Dezembro/1997.

GUPTA, Rayeev: *The "Glue" of Networks: Looking at IP over ATM;* Volume 7 Número 1; The ATM Forum Newsletter; ATM Forum: Fevereiro/1999.

ROMER, Paul: *In the beginning was the transistor;* History & Politics - FORBES. USA: 1996.

FAGGIN, Frederico: *The Future of the Microprocessor; Science & Technology* - Forbes. USA: 1996.

QDS: *Introduction to Objects;* http://www.qds.com/oo_intro/index.htm - USA: 1997.

BRISA: *Gerenciamento de redes: Uma abordagem de Sistemas Abertos* - Makron Books. Brasil: 1993.

GT P&D: *CG Internet/BR - Convite para Utilização de Plataformas de Alta Velocidade.* 1997.

ProTeM - RNP: *Edital – Projetos de Redes Metropolitanas de Alta Velocidade* - MEC. Brasília. BRASIL: 1998.

ProTeM - RNP: *Edital Complementar – Projetos de Redes Metropolitanas de Alta Velocidade* - MEC. Brasília. BRASIL: 1999.

ADVENTNET: *An introduction to SNMP: AdventNet Management Builder Documentation. Sao Jose, California.* http://www.adventnet.com/ mbuilder/help/index.html, 1998.

POP-RCT: *Site do Ponto de Presença da Rede de Ciência Tecnologia na UFSC.* Florianópolis, *Santa Catarina.* http://www.pop-ufsc.rct-sc.br. 1999.

POP-SC: *Site do Ponto de Presença da Rede Nacional de Pesquisa em Santa Catarina na UFSC.* Florianópolis, *Santa Catarina.* http://www.pop-sc.rnp.br. 1999.

UFSC: *Site* da Universidade Federal de Santa Catarina. Florianópolis, *Santa Catarina.* http://www.ufsc.br. 1999.

NPD: *Site do Núcleo de Processamento de Dados da UFSC.* Florianópolis, *Santa Catarina.* http://www.ufsc.br/npd. 1999.

FUNCITEC: *Site da Rede de Ciência e Tecnologia – FUNCITEC.* Florianópolis, *Santa Catarina.* http://www.funcitec.rct-sc.br. 1999.

REMAV-FLN: *Site da Rede Metropolitana de Florianópolis.* Florianópolis, *Santa Catarina.* http://www.rmav-fln.ufsc.br. 1999.

RNP: Site da Rede Nacional de Pesquisa. São Paulo, São Paulo. http://www.rnp.br. 1999.